Random Structures and Algorithms

Alan Frieze

Abstract. We provide an introduction to the analysis of random combinatorial structures and some of the associated computational problems.

Mathematics Subject Classification (2010). Primary 05C80; Secondary 68Q87.

Keywords. Random Graphs, Algorithms, Average Case.

1. Introduction

Our aim in this paper is to give a short survey on work on some probabilistic aspects of Combinatorics and their relation to algorithmic questions in Computer Science.

Combinatorics/Discrete Mathematics (in the main) concerns itself with certain properties of large, finite sets, with some defined structure.

Given such a set Ω, (often a set of graphs) we have certain natural questions:

1. How big is Ω: *Enumerative Combinatorics*.
2. How big is the greatest element of Ω: *Extremal Combinatorics*.
3. What are the properties of a typical member of Ω: *Probabilistic Combinatorics*.
4. What is the the complexity of computational problems associated with the above topics.

This paper will concern itself with Items 3. and 4. of this list. We should not confuse Item 3 with the *Probabilistic Method* where we use probabilistic notions to prove the existence of certain objects. We will try to interweave the structural analysis of random structures with related algorithmic questions. We refer the reader to Stanley [109] for Item 1. and to Bollobás [21] or Jukna [71] or Lovasz [90] for Item 2. The probabilistic analysis of algorithms has (at least) two flavors. Flavor 1 is a very detailed analysis of simple algorithms. See Sedgewick and Flajolet [106] for details on this. This paper will restrict attention to flavor 2; more complex algorithms for which the level of detail is less than that achieved in flavor 1.

We will begin with the seminal work of Erdős and Rényi on the evolution of random graphs. This is the subject matter of Section 2. We will then choose some topics as further illustration of the diverse aspects of the area. We have chosen Graph Coloring (Section 3); Matchings in random graphs (Section 4); Hamilton cycles (Section 5); and some questions about the optimal value of random optimization problems (Section 6).

■ Proceedings of the International Congress of Mathematicians, Seoul, 2014

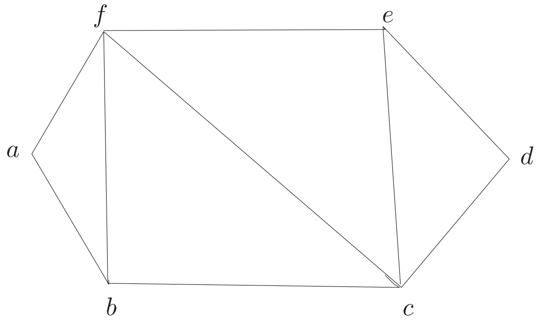
Our discussion of algorithms will focus on the probabilistic analysis of two NP-hard optimization problems. In the very early days of algorithmic analysis, it was considered sufficient to prove that an algorithm always terminated after a finite number of steps. This led to lexicographic versions of the simplex algorithm. When Integer Programming was realized to be important, Gomory's cutting plane algorithms [63] were considered to be a breakthrough. He showed that they would solve Integer Programming problems in finite time, but the bounds on running time were bad, and not stressed.

It was Edmonds [48] who pointed out in his seminal paper on matchings that "finite" can be very large and that we should try to find "good" algorithms i.e. those that always run in polynomial time. The search for polynomial time algorithms began in earnest. The most notable success in this quest being a polynomial time algorithm for Linear Programming, Khaciyan [80]. The optimism that most naturally occuring problems could be solved in polynomial time was crushed by the works of Cook [42], Levin [87] and Karp [74].

There have been several reactions to this negative state of affairs. One has been to see how well we can do in polynomial time, leading to the intense study of approximation algorithms. Another has been to focus on special cases that are quickly solvable. It should perhaps be pointed out that the cryptography community has turned this negative state of affairs into something positive.

To show that a problem can be difficult in the worst-case, one has to construct pathological examples. It is perhaps fortunate that in practice, the instances of NP-hard problems that are thrown up tend not to be pathological and large problems do get solved. Perhaps the best illustration of this comes from the success of researchers in solving large Traveling Salesperson Problems [9]. To explain this success, we have to define a "typical" problem and analyse the efficiency of algorithms on typical problems. For us, typical means drawn from some probability distribution and an algorithm will be considered efficient if is *expected* running time is polynomial in its input size or if it runs in polynomial time with high probability (w.h.p.)[1] We will describe some of the work in the area of analysing efficient algorithsm for random NP-hard problems. In particular, we will discuss the Traveling Salesperson Problem in Section 7 and Random k-SAT in Section 8.

1.1. Basic notions of graph theory. A graph $G = (V, E)$ consists of a set of vertices V together with a collection of edges $E \subseteq \binom{V}{2}$ i.e. E is a set of 2-element subsets of V. It is useful to imagine G as drawn below.



[1] A sequence of events $\mathcal{E}_n$ is said to occur w.h.p. if $\mathbf{Pr}(\mathcal{E}_n) = 1 - o(1))$ as $n \to \infty$.

The *degree* of a vertex is the number of edges that it lies in. A *walk* is a sequence $(w_1, w_2, \dots, w_k)$ where $\{w_{i-1}, w_i\} \in E$ for $1 \leq i < k$. A *path* is a walk in which $w_1, w_2, \dots, w_k$ are distinct. A *cycle* is a walk $w_1, w_2, \dots, w_k = w_1$ where $w_1, w_2, \dots, w_{k-1}$ are distinct. We can define an equivalence relation R on V where vRw iff G contains a path from v to w. The equivalence classes of this relation are called the *components* of G and a graph is *connected* if there is a unique component. Each component therefore forms a connected subgraph. A *tree* is a connected graph without cycles. If it has k vertices, then it necessarily has $k-1$ edges.

2. Evolution of a random graph

The analysis of random structures as a major field can be traced directly to the seminal paper [50] of Erdős and Rényi. Let us first establish notation.

One of the most important examples of a random structure is the random graph $G_{n,m}$, where $0 \leq m \leq \binom{n}{2}$. Here the vertex set is $[n] = \{1, 2, \dots, n\}$ and the edge set $E_{n,m}$ consists of m edges chosen uniformly at random. This is a subgraph of the complete graph $K_n = \left([n], \binom{[n]}{2}\right)$. The random graph $G_{n,m}$ is intimately related to the random graph $G_{n,p}$ where $0 \leq p \leq 1$. In this graph, also a subgraph of K_n, each edge of K_n is independently included as an edge with probability p. If $m \sim \binom{n}{2}p$ then $G_{n,p}$ and $G_{n,m}$ have "similar" properties.

The paper [50] describes the typical properties of $G_{n,m}$ for various values of m and as $n \to \infty$. The component structure of $G = G_{n,m}$ is summarised by the following.

(a) If $m = o(n^{1/2})$ then w.h.p. G consists of isolated edges and vertices.

(b) If $n^{\frac{k-1}{k}} \ll m = o(n^{\frac{k}{k+1}})$ then w.h.p. the components are trees with $1 \leq j \leq k+1$ vertices. Each possible such tree appears.

(c) If $m = cn$ for some constant $0 < c < \frac{1}{2}$ then almost all of the components are trees. There will be a few unicyclic components i.e. components containing a unique cycle. The maximum component size is $O(\log n)$.

(d) If $m \sim \frac{1}{2}n$ then the component structure is more complicated. The maximum component size is of order $n^{2/3}$. It has been the subject of intensive study e.g. Janson, Knuth, Łuczak and Pittel [69].

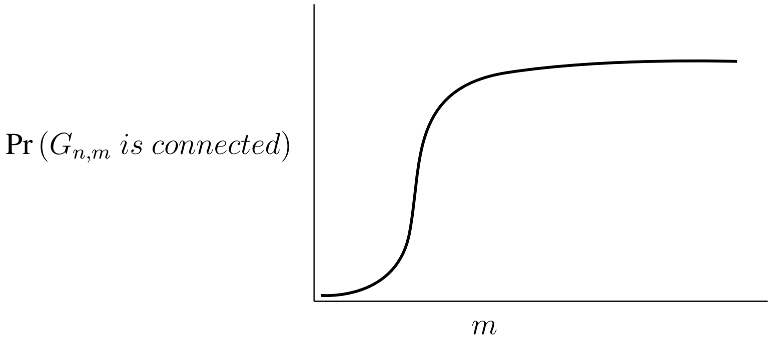
(e) If $m = \frac{1}{2}cn$ where $c > 1$ then w.h.p. there is a unique "giant" component of size $\sim \gamma(c)n$. The remaining components are almost all trees. The second largest component is of size $O(\log n)$. $\gamma(c)$ is the probability that a branching process where each particle has a Poisson, mean c, number of descendants, does not become extinct.

(f) If $m = \frac{1}{2}n(\log n + c_n)$ then [49],

$$\begin{aligned} \lim_{n\to\infty} \mathbf{Pr}(G_{n,m} \text{ is connected}) &= \begin{cases} 0 & c_n \to -\infty \\ e^{-e^{-c}} & c_n \to c \\ 1 & c_n \to +\infty \end{cases} \qquad (2.1) \\ &= \lim_{n\to\infty} \mathbf{Pr}(\delta(G_{n,m}) \geq 1) \end{aligned}$$

where δ denotes minimum degree.

Notice the sharp transition from being disconnected to connected, as claimed in (g). This is shown pictorially in the diagram below:



This leads us to the notion of *thresholds*. A function $\tau(n)$ is a threshold for a graph property $\mathcal{P}$ if

$$\lim_{n\to\infty} \mathbf{Pr}(G_{n,m} \text{ has } \mathcal{P}) = \begin{cases} 0 & \frac{m}{\tau(n)} \to 0 \\ 1 & \frac{m}{\tau(n)} \to \infty \end{cases}.$$

Thus $n \log n$ is the threshold for connectivity. Of course (2.1) claims something much stronger. What we have here is a *sharp* threshold. One of the main quests in the theory of random graphs is for the precise thresholds for important graph properties.

Before leaving this section, we remark on the proof technique introduced by Erdős and Rényi. There are $M = \binom{\binom{n}{2}}{m}$ distinct graphs with vertex set $[n]$ and m edges. The probability that $G_{n,m}$ is connected is then simply M_c/M where M_c is the number of connected graphs with vertex set $[n]$ and m edges. One gets nowhere if one tries to estimate the probability of connectivity by trying to evaluate M_c.

One key insight of Erdős and Rényi is to identify events that are very unlikely to occur. If one identfies the correct collection, then one can often reach the goal of estimating the probability of occurrence of the event that you are really interested in. As an example, we let $p = \frac{c \log n}{n}$ where $c > 1$ is constant. According to (2.1), we should be able to prove that $G_{n,p}$ is connected w.h.p. for this value of p.

Let X denote the number of components with at most $n/2$ vertices. We observe that a graph with n vertices is connected iff $X = 0$. Then

$$\mathbf{Pr}(X \neq 0) \leq \mathbf{E}(X) \leq \sum_{k=1}^{n/2} \binom{n}{k} k^{k-2} p^{k-1} (1-p)^{k(n-k)} \leq \frac{n}{c \log n} \sum_{k=1}^{n/2} \left(\frac{ce \log n}{n^{c(1-k/n)}} \right)^k \to 0. \tag{2.2}$$

Explanation: For a fixed k, there are $\binom{n}{k}$ choices for the vertex set of a component C. To ensure it is connected we choose a spanning tree T of C, in k^{k-2} ways. Then we multiply by the probablity p^{k-1} that T exists in $G_{n,p}$ and then by the probability $(1-p)^{k(n-k)}$ that there are no edges between C and the rest of the graph.

2.1. Hitting Times. Consider the sequence $G_0, G_1, \ldots, G_m, \ldots$, where G_{i+1} is G_i plus a random edge.

Let m_k denote the minimum m for which the minimum vertex degree $\delta(G_m) \geq k$. These are important "times" for the occurrence of important properties.

Theorem 2.1 (Erdős and Rényi [49]). *W.h.p. m_1 is the time when G_m first becomes connected.*

Observe that the largest term in the sum in (2.2) is for $k = 1$.

For other important properties, we need a couple of definitions. A *matching* in a graph G is a set of vertex disjoint edges. The matching is *perfect* if every vertex is covered by an edge of the matching. This is impossible if $|V|$ is odd, in which case we allow one uncovered vertex.

Theorem 2.2 (Erdős and Rényi [51]). *W.h.p. m_1 is the "time" when G_m first has a perfect matching.*

A *Hamilton cycle* in a graph G is a cycle that passes through each vertex exactly once.

Theorem 2.3 (Ajtai, Komlós, Szemerédi [4], Bollobás [20]). *W.h.p. m_2 is the time when G_m first has a Hamilton cycle.*

In general there will be many distinct Hamilton cycles at time τ_1.

Theorem 2.4 (Cooper and Frieze [43]). *W.h.p. at "time" m_2, G_m has $(\log n)^{n-o(n)}$ distinct Hamilton cycles.*

This was recently improved to

Theorem 2.5 (Glebov and Krivelevich [62]). *W.h.p. at time m_2, G_m has $n!p^n e^{-o(n)}$ distinct Hamilton cycles.*

One can also ask for edge disjoint Hamilton cycles. Let Property $\mathcal{A}_k$ denote the existence of $\lfloor k/2 \rfloor$ disjoint Hamilton cycles plus a disjoint perfect matching if k is odd.

Theorem 2.6 (Bollobás and Frieze [28]). *W.h.p. at time $m_k, k = O(1)$, G_m has property $\mathcal{A}_k$.*

We believed that the $k = O(1)$ bound was unnecessary. This has recently been verified.

Theorem 2.7 (Knox, Kühn and Osthus [82]). *W.h.p. G_m has property $\mathcal{A}_\delta$ for $n \log^{50} n \leq m \leq \binom{n}{2} - o(n^2)$.*

Theorem 2.8 (Krivelevich and Samotij [85]). *W.h.p. G_m has property $\mathcal{A}_\delta$ for $\frac{1}{2} n \log n \leq m \leq n^{1+\epsilon}$.*

3. Graph Coloring

A *proper* k-coloring of a graph $G = (V, E)$ is a map $f : V \to [k]$ such that if $\{v, w\}$ is an edge of G then $f(v) \neq f(w)$. The chromatic number $\chi(G)$ is the smallest k for which there is a proper k-coloring.

A set of vertices $S \subseteq V$ is *independent* if $v, w \in S$ implies that $\{v, w\}$ is not an edge. In a proper k-coloring, each color class is an independent set. The *independence number* $\alpha(G)$ is the size of a largest independent set.

3.1. Dense random Graphs.

Theorem 3.1 (Matula [93]). *W.h.p.*

$$\alpha(G_{n,1/2}) = 2\log_2 n - 2\log_2\log_2 n + O(1).$$

Finding an independent set of size $\sim \log_2 n$ in polynomial time is easy.

Greedy Algorithm:
Start with $I = \{1\}$.
Repeatedly add $v \in [n] \setminus (I \cup N(I))$ until no such v can be found.
(Here $N(I)$ is the set of neighbors of I i.e. $\{w \notin I : \exists v \in I\ s.t.\ \{v,w\} \in E\}$).

After k successful steps we find that the number of choices for v is distributed as the binomial
$Bin(n-k, 2^{-k})$. If $k \leq \log_2 n - 2\log_2\log_2 n$ then this is non-zero with probability $1 - o(1/\log n)$. So, w.h.p. the algorithm succeeds in finding an independent set of size at least $(1-o(1))\log_2 n$.

Surprisingly, no-one has been able to find a polynomial time algorithm that w.h.p. finds an independent set of size $(1+\epsilon)\log_2 n$ for any positive constant $\epsilon > 0$.

Indeed, it may not be possible to find such an independent set in polynomial time w.h.p. Deciding the truth of this is a challenging problem in complexity theory.

It follows from Matula's result that w.h.p. $\chi(G_{n,1/2}) \geq \frac{n}{2\log_2 n}$

Theorem 3.2 (Bollobás and Erdős [25], Grimmett and McDiarmid [64]). *W.h.p. a simple greedy algorithm uses* $\sim \frac{n}{\log_2 n}$ *colors.*

Given the fact that no-one knows how to find a large independent set in polynomial time, no-one knows how to find a coloring with at most $(1-\epsilon)n/\log_2 n$ colors in polynomial time.

It may even be NP-hard to find such a coloring in polynomial time w.h.p.

For a long time, no-one could prove an upper bound $\chi(G_{n,1/2}) \leq (1+o(1))\frac{n}{2\log_2 n}$.

The "discovery" of Martingale Concentration Inequalities was a great help. Let $Z = Z(X_1,\ldots,X_N)$ where $X_1,\ldots,X_N$ are independent. Suppose that changing one X_i only changes Z by ≤ 1. Then

$$\mathbf{Pr}(|Z - \mathbf{E}(Z)| \geq t) \leq e^{-2t^2/N}. \tag{3.1}$$

They were discovered by Shamir and Spencer [107] and by Rhee and Talagrand [99]. They have had a profound effect on the area. Further concentration inequalities by Talagrand [111] and Kim and Vu [81] have been extremely useful.

Concentration inequalities are extremely useful. They enable some random variables to be treated more or less like constants. The inequality (3.1) is a special case of what has become known as the Azuma-Hoeffding concentration inequality.

Theorem 3.3 (Bollobás [22]). $\chi(G_{n,1/2}) \sim \frac{n}{2\log_2 n}$.

Proof. Let Z be the maximum number of independent sets in a collection $S_1,\ldots,S_Z$, $|S_i| \sim 2\log_2 n$ and $|S_i \cap S_j| \leq 1$.

$$\mathbf{E}(Z) = n^{2-o(1)}\ \textit{and changing one edge changes Z by}\ \leq 1$$

So,

$$\mathbf{Pr}(\exists S \subseteq [n] : \ |S| \geq \frac{n}{(\log_2 n)^2} \ \mathit{and}\ S\ \mathit{doesn't\ contain\ a}$$
$$(2 - o(1)) \log_2 n \ \mathit{independent\ set}) \leq 2^n e^{-n^{2-o(1)}} = o(1).$$

So, we color $G_{n,1/2}$ with color classes of size $\sim \ 2\log_2 n$ until there are $\leq \ n/(\log_2 n)^2$ vertices uncolored and then give each remaining vertex a new color. □

3.2. Sparse Random Graphs. There has recently been a lot of research concerning the chromatic number of sparse random graphs viz. $G_{n,p}$, $p = d/n$ where $d = O(1)$.

Conjecture. *There exists a sequence $d_k : k \geq 2$ such that w.h.p.*

$$\chi(G_{n,d/n}) = k \, \mathit{for}\ d_{k-1} < d < d_k.$$

Friedgut [53] and Achlioptas and Friedgut [1] came close to proving this. Friedgut [53] (with an appendix by Bourgain) characterised properties which had a sharp threshold in terms of the non-existence of small local obstructions. He showed that if there are no small obstructions then for each n there is a value τ_n such that the property is likely to occur close to τ_n. Unfortunately, at the moment there is no general proof that the sequence (τ_n) tends to a limit. Friedgut's prime example was for k-SAT, described later, and together with Achlioptas, he showed the existence of such a sequence (τ_n) in the case of k-colorability.

It was soon established that w.h.p. the chromatic number only took one of two values:

Theorem 3.4 (Łuczak [91]). *W.h.p. $\chi(G_{n,d/n})$ takes one of two values.*

Surprisingly, using Chebyshev's inequality we get

Theorem 3.5 (Achlioptas and Naor [2]). *Let k_d be the smallest integer $k \geq 2$ such that $d < 2k \log k$ then w.h.p. $\chi(G_{n,d/n}) \in \{k_d, k_d + 1\}$.*

We find this surprising as we would have expected the variance of the number of k-colorongs to be too large to apply.

If X denotes the number of k-colorings of $G_{n,d/n}$ then

$$\mathbf{Pr}(X > 0) \geq \frac{\mathbf{E}(X)^2}{\mathbf{E}(X^2)} = \Omega(1)$$

for $d < (k-1)\log(k-1)$.
This shows that $\mathbf{Pr}(X > 0)$ is bounded below by a constant. We can now use Achlioptas and Friedgut.

The idea is straightforward. The difficulty lies in estimating the ratio $\mathbf{E}(X)^2/\mathbf{E}(X^2)$.

Achlioptas and Naor showed that for approximately half of the possible values for d, $\chi(G_{n,d/n})$ is determined w.hp.

Theorem 3.6 (Achlioptas and Naor [2]). *If $d \in ((2k-1)\log k, 2k \log k)$ then w.h.p.*

$$\chi(G_{n,d/n}) = k + 1.$$

This has been improved so that we now have

Theorem 3.7 (Coja-Oghlan and Vilenchik [40]). *Let κ_d be the smallest integer $k \geq 2$ such that $d < (2k-1)\log k$. Then $\chi(G_{n,d/n}) = \kappa_d$ for $d \in \mathcal{A}$ where $\mathcal{A}$ has density one in R_+.*

Furthermore, Coja-Oghlan has also improved on the naive first moment upper bound.

Theorem 3.8 (Coja-Oghlan [35]).

$$d_k \leq 2k \log k - \log k - 1 + o_k(1).$$

Now for large k, the value of d_k is known within an interval of length less than 0.39.

There is still a factor of two gap between what can be proved existentially and what can be proved to be constructible in polynomial time.

4. Matchings

The seminal paper of Edmonds [48] showed that a matching of maximum size in a graph can be found in polynomial time. The algorithm is relatively complicated. Karp and Sipser [78] proposed the following greedy algorithm for finding a large matching:

KSGREEDY

```
begin
    $M \leftarrow \emptyset$;
    while $E(G) \neq \emptyset$ do
    begin
        A1: If $G$ has a vertex of degree one, choose one, $x$ say, randomly.
            Let $e = \{x, y\}$ be the unique edge of $G$ incident with $x$ Endif;
        A2: Else, (no vertices of degree one) choose
            $e = \{x, y\} \in E$ randomly Endelse;

            $G \leftarrow G \setminus \{x, y\}$;
            $M \leftarrow M \cup \{e\}$
    end;
    Output $M$
end
```

Note that this algorithm never makes a ‘mistake’ when executing command **A1**: If the graph G has a degree 1 vertex x then there is a maximum matching that contains the unique edge that contains x.

Karp and Sipser analysed the algorithms performance on $G_{n,p}$, where $p = c/n$. The random graph $G_{n,p}$ with $p = c/n$ will have a linear number of vertices of degree 1 w.h.p. The Karp Sipser Algorithm will therefore have an initial phase, which we call Phase 1, in which it executes command A1 in every step. After all degree 1 vertices are exhausted (for the first time) we move to Phase 2. During this phase both A1 and A2 are performed.

Theorem 4.1 (Karp and Sipser [78]). *If $c < e$ then w.h.p. Phase 1 ends with a graph with $o(n)$ vertices. If $c \geq e$ then w.h.p. Phase 2 isolates $o(n)$ vertices.*

Theorem 4.2 (Aronson, Frieze and Pittel [10]). *If $c < e$ then w.h.p. Phase 1 ends with a graph consisting of a few vertex disjoint cycles. (The expected number of vertices on these cycles is $O(1)$). If $c > e$ then w.h.p. Phase 2 isolates $\Theta(n^{1/5} \log^{O(1)} n)$ vertices.*

Sketch of Proof. The proof of Theorem 4.2 illustrates the use of differential equations in the analysis of algorithms. We now sketch a proof.

For the graph G_i remaining after i steps of the algorithm, let

$$\begin{aligned} \mathbf{v}_1 &= \text{the number of vertices of degree one} \\ \mathbf{v} &= \text{the number of vertices of degree at least two} \\ \mathbf{m} &= \text{the number of edges} \end{aligned}$$

It is not hard to show that if we condition on the values of $\mathbf{v}, \mathbf{v}_1$ and $\mathbf{m}$ then G is uniformly distributed on the set of graphs with these parameters. We use this fact to determine the likely evolution of the remaining graph. We can show that $\mathbf{v}, \mathbf{v}_1$ and $\mathbf{m}$ are very likely to follow their expected trajectories. These trajectories are given by the solution of a set of differential equations in variables, v, v_1, m.

Let $\mathbf{v}_k$ be the number of vertices of degree k in G_i. One can show that $\mathbf{v}_k \approx v_k$ where

$$v_k = \frac{vz^k}{k!(e^z - 1 - z)}$$

where z is the solution to

$$\frac{2m - v_1}{v} = \frac{z(e^z - 1)}{f}, \qquad f = f(z) = e^z - 1 - z.$$

The differential equations arise from the consideration of the expected change in these parameters in one step.

One step transitions: If v_1', v', m' denote the values of the parameters after one step of the algorithm then, given v_1, v, m

$$\begin{aligned} \mathbf{E}[\mathbf{v}_1' - \mathbf{v}_1] &\approx -1 - \frac{\mathbf{v}_1}{2\mathbf{m}} + \frac{\mathbf{v}^2 z^4 e^z}{(2\mathbf{m}f)^2} - \frac{\mathbf{v}_1 \mathbf{v} z^2 e^z}{(2\mathbf{m})^2 f}, \\ \mathbf{E}[\mathbf{v}' - \mathbf{v}] &\approx -1 + \frac{\mathbf{v}_1}{2\mathbf{m}} - \frac{\mathbf{v}^2 z^4 e^z}{(2\mathbf{m}f)^2}, \\ \mathbf{E}[\mathbf{m}' - \mathbf{m}] &\approx -1 - \frac{vz^2 e^z}{2\mathbf{m}f}. \end{aligned}$$

$\mathbf{v}_1, \mathbf{v}, \mathbf{m}$ closely follow the trajectory of a set of differential equations. These equations model the one step transitions.

$$\frac{dv_1}{dt} = -1 - \frac{v_1}{2m} + \frac{v^2 z^4 e^z}{(2mf)^2} - \frac{v_1 v z^2 e^z}{(2m)^2 f}; \quad \frac{dv}{dt} = -1 + \frac{v_1}{2m} - \frac{v^2 z^4 e^z}{(2mf)^2}; \quad \frac{dm}{dt} = -1 - \frac{vz^2 e^z}{2mf}.$$

Their solution is, where $\beta e^{c\beta} = e^z$,

$$2m = \frac{n}{c} z^2; \tag{4.1}$$

$$\begin{aligned} v =&n(1-e^{-z}(1+z))\beta; \\ v_1 =&\frac{n}{c}\left[z^2-zc\beta(1-e^{-z})\right]; \\ t =&\frac{n}{c}\left[c(1-\beta)-\frac{1}{2}\log^2\beta\right]. \end{aligned}$$

Sub-critical case: $c < e$

Let $h(z) = \frac{n}{c}\left[z^2-zc\beta(1-e^{-z})\right]$. Let z^* be the largest nonnegative root of $h(z) = 0$. If $c < e$ then $z^* = 0$. Because the process closely follows (4.1) we see that $z = o(1)$ w.h.p. at the end of Phase 1. This gives $m = o(n)$ (actually $m = o(n^{0.9})$) and then a careful first moment calculation yields the conclusion of Theorem 4.2.

Super-critical case: $c > e$

In this case we end Phase 1 with $z = z^* > 0$.

We have observed that

$$\mathbf{E}[v_0' - v_0] = O\left(\frac{v_1}{m}\right) \text{ — expected increase in unmatched vertices.}$$

It is enough to show that w.h.p. $v_1 = \tilde{O}(n^{1/5})$ throughout the algorithm. Then we can argue that w.h.p. there are

$$\tilde{O}\left(n^{1/5}\sum_{m=1}^{cn}\frac{1}{m}\right)$$

vertices left unmatched in Phase 2.

Controlling v_1: We first observe that

$$v_1 > 0 \text{ implies } \mathbf{E}[v_1' - v_1] \leq -\min\left(\frac{z^2}{200}, \frac{1}{20000}\right)$$

Early Phase: $z \geq n^{-1/100}$.

Whp v_1 stays $\tilde{O}(z^{-2})$.

Middle Phase: $n^{-1/100} \geq z \geq n^{-1/5}$

The graph is very sparse, most vertices are of degree two.

When $v_1 > 0$ most vertices of degree one are at end of a long path. Removing such a vertex and its edge does not change v_1 i.e.

$$\mathbf{Pr}(v_1' = v_1 \mid v_1 > 0) = 1 - z + O(z^2).$$

Whp v_1 stays $\tilde{O}(z^{-1})$.

Final Phase: $z \leq n^{-1/5}$

We start this phase with

$$v \sim v_2 \sim Cnz^2 = \tilde{O}(n^{3/5})$$

Only $\tilde{O}(n^{3/5}z) = \tilde{O}(n^{2/5})$ moves made in the "v_1 walk" and so v_1 can only move by the square root of this.

Thhis completes our outline analysis of the Karp-Sipser algorithm. □

The Karp-Sipser algorithm runs in $O(n)$ time and makes $\tilde{O}(n^{1/5})$ mistakes.

Chebolu, Frieze and Melsted [31] show that these mistakes can be corrected i.e one can find a true maximum matching in $O(n)$ time w.h.p., for c sufficiently large.

The key idea of this paper is that even though we have "looked" at all of the edges while running KSGREEDY, there is enough "residual randomness" to use to find a true maximum matching. We have to examine the output of the algorithm for unused edges whose distribution can be properly understood.

5. Hamilton Cycles

Determining whether or not a graph has a Hamilton cycle is NP-hard, Karp [74].

The threshold problem was solved in

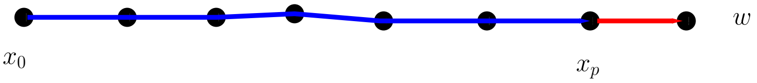
Theorem 5.1 (Komlós and Szemerédi [83]). *Suppose that* $m = \frac{1}{2}n(\log n + \log\log n + c_n)$. *Then*

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

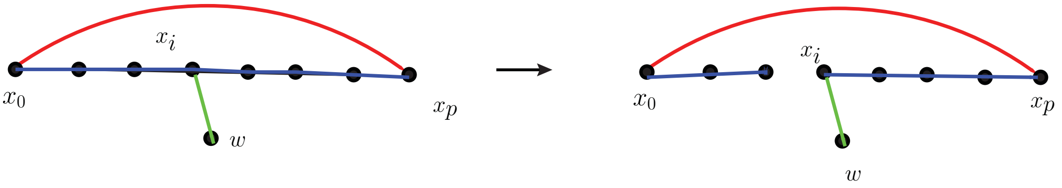
We will describe an algorithm that runs in polynomial time and finds a Hamilton cycle w.h.p. for the case $c_n = \omega \to \infty$.

5.1. Pósa Rotations. We can start our algorithm with any path $P = (x_0, x_1, \ldots, x_p)$. Each round of the algorithm tries to replace P by a path Q of length one more i.e. $p + 1$. If successful, Q replaces P for the next round. This continues until we have a path of length $n - 1$ i.e. a Hamilton path. After this, we attempt to create a Hamilton cycle.

If there is an edge joining x_0 or x_p to a vertex w not in P, then we can extend the path to a longer one. With the edge $\{x_p, w\}$ we have the longer path $Q = (x_0, x_1, \ldots, x_p, w)$. We call this a *simple* extension.

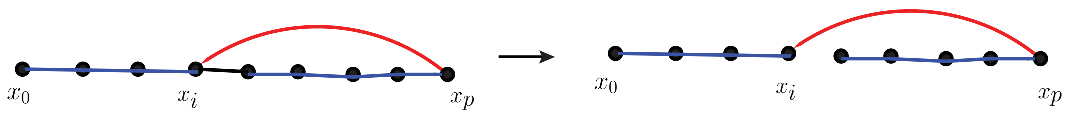


There is an alternative way of extending a path:



If the edge $\{x_0, x_p\}$ exists then so will the edge $\{x_i, w\}$, for some i, and for some $w \notin P$, unless $P + \{x_0, x_p\}$ is a hamilton cycle. This is under the assumption that our graph is connected. We then have the longer path $(w, x_i, x_{i+1}, \ldots, x_p, x_0, \ldots, x_{i-1})$. We call this a *cyclic* extension.

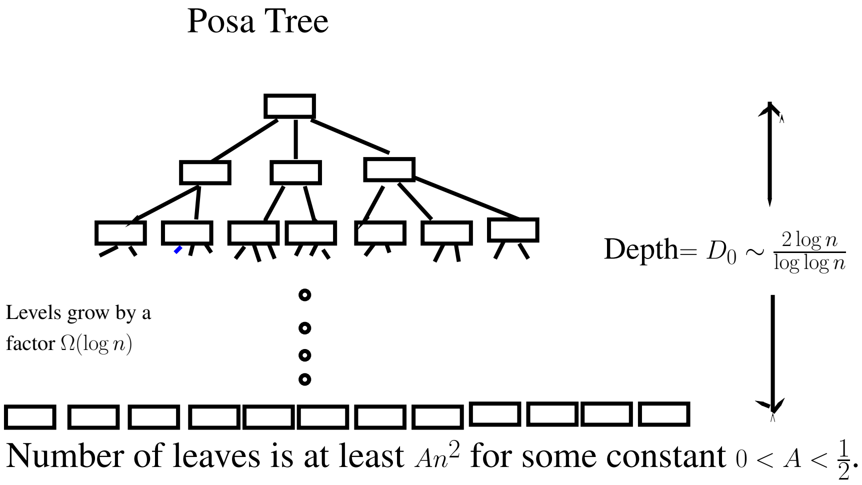
If there is no extension of either type, then we *rotate* the path:



Given the edge $\{x_p, x_i\}$ where $0 < i < p-1$, we create the new path $(x_0, \ldots, x_i, x_p, x_{p-1}, \ldots, x_{x+1})$, of the same length as P. By doing this we gain a new opportunity for an extension.

Pósa [98] used these rotations in a breakthrough paper that found the threshold for Hamiltonicity. He showed that if $m \geq Kn \log n$ for some constant $K > 0$ then $G_{n,m}$ is hamiltonian w.h.p.

Let $m = \frac{1}{2}n(\log n + \log\log n + \omega)$ and $m_2 = \omega n/2$ and let $m_1 = m - m_2$. In the diagram below, labeled "Posa Tree", each rectangle R represents a path $P(R)$ of the same length as the path represented by the root rectangle. Rectangle R_1 is the child of rectangle R_0 iff $P(R_1)$ is obtained from $P(R_0)$ by a single rotation.


Posa Tree
Depth$= D_0 \sim \frac{2\log n}{\log\log n}$
Levels grow by a factor $\Omega(\log n)$
Number of leaves is at least An^2 for some constant $0 < A < \frac{1}{2}$.

The algorithm we have in mind can be described as follows: At the beginning of a round we have a path P. We build a Pósa tree as indicated in the diagram, only using the first m_1 edges. If at any stage we can extend one of the paths that we have constructed, then we extend it and go to the next round. Otherwise, we can show that w.h.p. we will end up with at least An^2 paths, each path having a distinct pair of endpoints. Here A is a positive constant. We denote the set of pairs of endpoints by BOOST. We then go to our m_2 edges and check one by one to see if any of them lies in BOOST. If so, we can make a cycle extension and move to the next round. We have enough edges so that we need not use an edge in more than one round.

The probability this algorithm fails can therefore be bounded by

$$\mathbf{Pr}(Bin(\omega_2 n, A) < n) = o(1).$$

It is polynomial time, because w.h.p. the Pósa tree has an $O(\log n)$ branching factor at each vertex.

This algorithm has an unintuitive feel. Why should we restrict the roles of the edges to either tree building or cycle closing? We now describe the algorithm of Bollobás, Fenner and Frieze [26] which is similar to what we have described, but makes no partition of the edges and furthermore is deterministic. It proceeds in rounds and in each round it uses all of the m available edges.

In a successful construction of a Pósa tree there are $O(\log n/\log\log n)$ edges that are "vital". These are the edges that are involved in the successful sequence of rotations that lead to an extension.

Let W be the set of vital edges picked out this way in all successful rotations, or incident to vertices of "low" degree. Then $|W| = O(n\log n/\log\log n)$.

We organise our algorithm deterministically so that the following is true. If the algorithm fails on G and we delete a set of edges X from G, where

$$\text{(i) } X \cap W = \emptyset \text{ and (ii) } X \text{ forms a matching}$$

then re-running the algorithm on $G - X$ leads to failure at exactly the same stage.

Suppose $|X| = \omega = \log n$ and that after removing X, any Posa tree has at least An^2 leaves. This is true w.h.p.

Consider the $\{0,1\}$ function $\psi(M,X)$ where M ranges over $\binom{\binom{n}{2}}{m}$ and X ranges over $\binom{M}{\omega}$.

$\psi(M,X) = 1$ iff X satisfies (i), (ii) and our algorithm fails on the graph $G = ([n], M)$.

Observe that

$$\sum_X \psi(M,X) > 0 \text{ implies } \sum_X \psi(M,X) \geq \binom{m}{\omega} \times \left(1 - O\left(\frac{\log\log n}{\log n}\right)^{\omega}\right).$$

So,

$$\mathbf{Pr}(\text{Algorithm fails}) \leq \frac{\sum_{M,X} \psi(M,X)}{\binom{m}{\omega} \times \left(1 - O\left(\frac{\log\log n}{\log n}\right)\right)^{\omega}} \times \frac{1}{\binom{N}{m}}$$

where $N = \binom{n}{2}$.

But,

$$\sum_{M,X} \psi(M,X) \leq \binom{N}{m-\omega}\binom{N - m + \omega - An^2}{\omega} \leq \binom{N}{m}\binom{m}{\omega}(1-A)^{\omega}.$$

This is because having chosen $M \setminus X$ in $\binom{N}{m-\omega}$ ways, we can't choose an edge that lies in BOOST, if we want to have $\psi(M,X) = 1$.

Therefore

$$\mathbf{Pr}(\text{Algorithm fails}) \leq \frac{(1-A)^{\omega}}{\left(1 - O\left(\frac{\log\log n}{\log n}\right)\right)^{\omega}} = o(1).$$

□

5.2. Sparse random graphs. With the threshold problem solved, existentially and constructively, we can consider other models of a random graph. In particular to those models where a minimum degree condition is automatically satisfied.

Let $G(n,r)$ denote a random r-regular graph chosen uniformly from the set of all graphs with vertex set $[n]$. (Regular means that all vertices have the same degree)

Theorem 5.2.

$$\lim_{n\to\infty} \mathbf{Pr}(G(n,r) \textit{ is Hamiltonian}) = 1, \quad r \geq 3.$$

$r = O(1)$ was proved by Robinson and Wormald [101], [102].
$r \to \infty$ was proved by Krivelevich, Sudakov, Vu, Wormald [86] and Cooper, Frieze, Reed [46].

If each vertex independently chooses k random neighbors then we have the random graph G_{k-out}.

Theorem 5.3 (Bohman and Frieze [19]).

$$\lim_{n\to\infty} \mathbf{Pr}(G_{k-out} \textit{ is Hamiltonian}) = 1, \quad k \geq 3.$$

This is not implied by the previous results on random regular graphs.

Let $G_{n,m;k}$ be sampled uniformly from all graphs with vertex set $[n]$ that have m edges and minimum degree at least k. In this way the minimum degree condition is obtained directly by conditioning.

Theorem 5.4 (Bollobás, Fenner and Frieze [27]). *Let* $m = \frac{1}{6}n(\log n + \log\log n + c_n)$ *then*

$$\lim_{n\to\infty} \mathbf{Pr}(G_{n,m;2} \textit{ is Hamiltonian}) = \begin{cases} 0 & c_n \to -\infty \\ e^{-f(c)} & c_n \to c \\ 1 & c_n \to +\infty \end{cases}$$

for some explicit function $f(c)$.

Now consider conditioning on minimum degree at least 3. Let

$$L_c = \lim_{n\to\infty} \mathbf{Pr}(G_{n,cn;3} \text{ is Hamiltonian})$$

Conjecture: $L_c = 1$ for all $c \geq 3/2$.

The conjecture is true for $c = 3/2$. In this case we are dealing with random 3-regular graphs.

Theorem 5.5 (Bollobás, Cooper, Fenner, Frieze [24]). $L_c = 1$ *for* $c \geq 128$.

Theorem 5.6 (Frieze [55]). $L_c = 1$ *for* $c \geq 10$.

The conjecture is true for $c \geq 3$, assuming a numerical solution of some differential equations.

Sketch of Proof. The proof for the case $c \geq 10$ is based on the analysis of a greedy algorithm for finding a good 2-matching in the random graph $G_{n,cn;3}$. A 2-matching is a set of edges M where no vertex meets more than two edges of M (as opposed to one for a matching). By "good" we mean that as a spanning subgraph of $[n]$, it has $O(\log n)$ components. The greedy algorithm can be thought of as a natural generalisation of the Karp-Sipser algorithm for matchings.

Given M, one can convert it to a Hamilton cycle, relatively easily. This being based on the relatively few components that it has. □

Following the approach of [31] to finding a perfect matching, Frieze and Haber [56] devised a fast algorithm to find the promised Hamilton cycle. It is based on first finding a good 2-matching and uses the "residual randomness" left by the greedy algorithm.

Theorem 5.7 (Frieze and Haber [56]). *If c is sufficiently large then w.h.p. a hamilton cycle can be found in $G_{n,cn;3}$ in $O(n^{1+o(1)})$ time.*

6. Edge Weighted Graphs

In this section we consider some results for the expected optimal objective value of some classical well solved problems in Combinatorial Optimization when the costs are random.

6.1. Spanning Trees. Every edge e of the complete graph K_n is given a random length X_e. The edge lengths are independently uniform $[0, 1]$ distributed. Z_n is the minimum total length of a spanning tree. A spanning tree is a connected subgraph of K_n with no cycles. It has $n - 1$ edges.

The conceptually simplest algorithm for finding a minimum spanning tree in a an edge weighted graph $G = (V, E)$ is the greedy algorithm. We order the edges of G as $e_1, e_2, \ldots, e_m$ where $\ell(e_i) \leq \ell(e_{i+1})$, here ℓ denotes edge length. Let $G_i = (V, \{e_1, e_2, \ldots, e_i\}$.

GREEDY

```
    $I \leftarrow \emptyset$.
    For $i = 1, 2, \ldots, m$ do
        begin
            If $e_i$ joins two disitinct components of $G_{i-1}$ then $I \leftarrow I \cup \{e_i\}$.
        end;
    Output $I$
    end
```

The property we need from this is the following. Fix p and let $G_p = (V, E_p)$ where
$E_p = \{e \in E : \ell(e) \leq p\}$ and let $I_p = I \cap E_p$. Suppose that the graph G_p has $=\!\!{}_{\varsigma}(G_p)$ components. Then $|I \setminus I_p| = {}_{\varsigma}1$.

Let T be the minimum spanning tree. In the following, G_p has the same distribution as $G_{n,p}$. Then,

$$
\begin{aligned}
Z_n = \ell(T) &= \sum_{e \in T} X_e \\
&= \sum_{e \in T} \int_{p=0}^{1} 1_{(p < X_e)} dp \qquad (6.1) \\
&= \int_{p=0}^{1} \sum_{e \in T} 1_{(p < X_e)} dp \\
&= \int_{p=0}^{1} |\{e \in T : p < X_e\}| dp \\
&= \int_{p=0}^{1} ({}_{\varsigma}(G_p) - 1) dp.
\end{aligned}
$$

Equation (6.1) follows from $x = \int_{p=0}^{1} 1_{(p \leq x)} dp$ for any $x \in [0,1]$.

Fact. $p \geq 6 \log n/n$ implies that $G_{n,p}$ is connected with sufficiently high probability.

Fact. Almost all of the integral is accounted for by small isolated tree components.

So,

$$\begin{aligned}
\mathbf{E}(Z_n) &= \int_{p=0}^{1} (\mathbf{E}((G_p)) - 1) dp & (6.2)\\
&\sim \int_{p=0}^{6 \log n/n} \mathbf{E}(\#\text{ small isolated trees in } G_{n,p}) dp \\
&\sim \int_{p=0}^{6 \log n/n} \left(\sum_{k=1}^{\log^2 n} \binom{n}{k} k^{k-2} p^{k-1} (1-p)^{k(n-k)+\binom{k}{2}-k+1} \right) dp & (6.3)\\
&\sim \sum_{k=1}^{\log^2 n} \frac{n^k}{k!} k^{k-2} \frac{k!(k(n-k)!}{(k(n-k+1)!} \\
&\sim \sum_{k=1}^{\log^2 n} \frac{1}{k^3} \\
&\sim \zeta(3).
\end{aligned}$$

Expanation for (6.3)**:** We choose the vertices S of the small tree in $\binom{n}{k}$ ways. We choose a spanning tree T of S in k^{k-2} ways. The probability that T exists is p^{k-1} and the probability that there are no other edges in S is $(1-p)^{\binom{k}{2}-k+1}$ and the probability that T is not connected to the rest of G_p is $(1-p)^{k(n-k)}$.

The above is most of the proof of the following:

Theorem 6.1 (Frieze [54]).

$$Z_n \sim \zeta(3) \quad w.h.p.$$

□

The original proof was not so "clean": The remarkable integral formula is due to Janson [67].

With more work we have

Theorem 6.2 (Cooper, Frieze, Ince, Janson, Spencer [45]).

$$\mathbf{E}(Z_n) = \zeta(3) + \frac{c_1}{n} + \frac{c_2 + o(1)}{n^{4/3}}.$$

The constants c_1, c_2 are made explicit in [45], but they are not "pretty".

If we give random weights to an arbitrary r-regular graph G then under some mild expansion assumptions

Theorem 6.3 (Beveridge, Frieze, McDiarmid [17]).

$$Z_n \sim \frac{n}{r}(\zeta(3) + \epsilon_r) \quad w.h.p.$$

where $\epsilon_r \to 0$ *as* $r \to \infty$.

For example, if G is the complete bipartite graph $K_{n/2,n/2}$ then $Z_n \sim 2\zeta(3)$ w.h.p.

6.2. Shortest Paths. Here we consider the following problem. Every edge e of the complete graph K_n is given a random length X_e. The edge lengths are independently distributed with an exponential distribution, mean one, i.e. $\mathbf{Pr}(X_e \geq t) = e^{-t}$ for all $t \geq 0$.

We let $D(i,j)$ denote the shortest distance between vertex i and j i.e. the minimum length of a path from i to j.

Theorem 6.4 (Janson [68]). *Let $D_{i,j}$ be the shortest distance between i,j in the above model. Then*

$$D_{1,2} \sim \frac{\log n}{n}; \quad \max_j D_{1,j} \sim \frac{2\log n}{n}; \quad \max_{i,j} D_{i,j} \sim \frac{3\log n}{n}.$$

The proof of this is based on an analysis of the well know Dijkstra algorithm for finding shortest paths from a fixed vertex s to all other vertices in a non-negatively edge weighted graph.

After several iterations of this algorithm there is a tree T, rooted at s, such that if v is a vertex of T then the tree path from s to v is a shortest path. Let $d(v)$ be its length. For $x \notin T$ let $d(x)$ be the minimum length of a path P that goes from s to v to x where $v \in T$ and the sub-path of P that goes to v is the tree path from s to v. Then if $d(y) = \min\{d(x) : x \notin T\}$ then $d(y)$ is the length of a shortest path from s to y and y can be added to the tree.

Suppose that vertices are added to the tree in the order $v_1, v_2, \ldots, v_n$ and that $Y_j = dist(v_1, v_j)$ for $j = 1, 2, \ldots, n$. It follows from the memoryless property of the exponential distribution that

$$Y_{k+1} = \min_{\substack{i=1,2,\ldots,k \\ j=k+1,\ldots,n}} [Y_i + X_{v_i,v_j}] = Y_k + E_k$$

where E_k is exponential with rate $k(n-k)$ and is independent of Y_k. This is because $X_{\{v_i,v_j\}}$ is distributed as an independent exponential X conditioned on $X \geq Y_k - Y_i$.

Hence

$$\mathbf{E}Y_n = \sum_{k=1}^{n-1} \frac{1}{k(n-k)} = \frac{1}{n}\sum_{k=1}^{n-1}\left(\frac{1}{k} + \frac{1}{n-k}\right) = \frac{2}{n}\sum_{k=1}^{n-1}\frac{1}{k} = \frac{2\log n}{n} + O(n^{-1}).$$

The Chebyshev inequality can be used to show concentration around the mean. This yields the second part of Theorem 6.4. The first part comes from the fact that vertex 2 is on average the $\sim n/2$th vertex added to the tree.

6.3. Assignment Problem. The assignment problem is the name given to the problem of finding a minimum weight perfect matching in a complete bipartite graph $K_{n,n}$ where each edge is given a weight. ($K_{n,n}$ has vertices V_1, V_2 where V_1, V_2 can be thought of as disjoint copies of $[n]$. There is an edge $\{i,j\}$ for every $i \in V_1, j \in V_2$.)

In the random assignment problem, each edge e of $K_{n,n}$ is given an independent random edge weight from some distribution $\mathcal{D}$. Let A_n denote the expected value of the minimum weight matching. Walkup [113] proved that $\mathbf{E}(A_n) \leq 3$ when $\mathcal{D}$ is the uniform $[0,1]$ distribution. Karp [77] improved this to $\mathbf{E}(A_n) \leq 2$. At this point there was no proof that $\lim_{n\to\infty}\mathbf{E}(A_n)$ existed or not. Aldous [5] proved the limit existed and in a follow up [6] proved that $\lim_{n\to\infty}\mathbf{E}(A_n) = \zeta(2) = \frac{\pi^2}{6}$. Here the distribution $\mathcal{D}$ is the exponential with mean one. Parisi [97] conjectured that the following is true.

$$\mathbf{E}(A_n) = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \cdots + \frac{1}{n^2}. \tag{6.4}$$

Theorem 6.5 (Linusson and Wästlund [89], Nair, Prabhakar and Sharma [94]). *Equation* (6.4) *holds.*

It took quite a deal of effort to prove this theorem. Wástlund however, [114] finally gave a short proof of it.

7. Traveling Salesperson Problem- TSP

Having seen some results about polynomially time solvable problems. We now discuss two important NP-hard problmes. We begin with the TSP. This is important in the history of the average case analysis of algorithms in that the paper by Karp [75] was an influential paper that proposed average case analysis as an antidote to the negative consequences of NP-completeness.

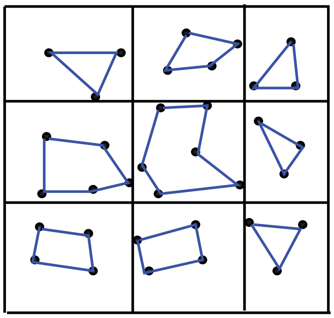
7.1. Euclidean Version. We begin with the Euclidean version of the TSP. Let $\mathcal{X} = \{X_1, X_2, \dots, X_n\}$ be chosen independently and uniformly from $[0,1]^2$.

Theorem 7.1 (Beardwood, Halton and Hammersley [13]). *There exists an absolute constant* $\beta > 0$ *such*

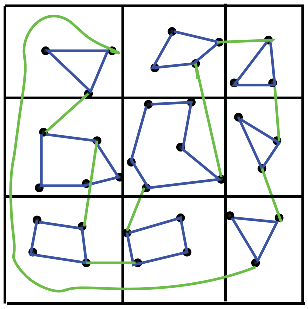
$$\frac{Z}{n^{1/2}} \to \beta \textit{ with probability 1}$$

The precise value of β is unkown to this day.

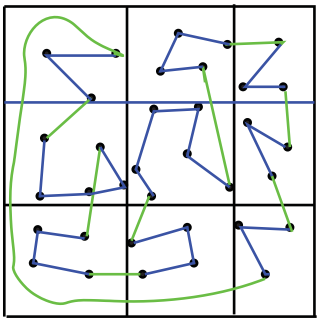
Karp [75] described a heuristic that runs in polynomial time and w.h.p. finds a tour of length within $o(n^{1/2})$ of the minimum. Here is a simplified version of Karp's algorithm. First divide the unit square up into $n/\log n$ subsquares of side $(\log n/n)^{1/2}$. Each subsquare will w.h.p. contain $O(\log n)$ points of $\mathcal{X}$. We can then use Dynamic Programming [66] to solve these problems in polynomial time.



Solve the individual problems in each sub-square.



Connect up the smaller tours as shown



Now remove edges to create a tour as shown in the diagram

It is not difficult to prove that the difference between the tour found by this heuristic and the optimal tour is bounded by a constant times the total length of the lines used to create the grid i.e. $O((n/\log n)^{1/2})$.

7.2. Independent Costs Version. We are given an $n \times n$ matrix $[c_{i,j}]$ where we assume that the $c_{i,j}$ are independent uniform $[0,1]$ variables.

The aim is to compute

$$T(C) = \min \left\{ \sum_{i=1}^{n} c_{i,\pi(i)} \ : \ \pi \text{ is a } \textbf{cyclic} \text{ permutation of } [n] \right\}$$

We compare this with the Assignment Problem already discussed. This can be re-formulated as to compute

$$A(C) = \min \left\{ \sum_{i=1}^{n} c_{i,\pi(i)} \ : \ \pi \text{ is a permutation of } [n] \right\}.$$

Here

$$\frac{\pi^2}{6} \sim A(C) \leq T(C) \leq A(C) + o(1) \ w.h.p.$$

The RHS is due to Karp [76].

The TSP can then be thought of as finding a minimum weight cycle cover of the complete digraph in which there is only one cycle. Here a cycle cover is a set of vertex disjoint oriented cycles that cover every vertex.

Karp's Patching Algorithm:

- Solve the associated assignment problem.
- *Patch* the cycles together to get a tour. Karp observed that if C is a matrix with i.i.d. costs then the optimal permutation is uniformly distributed and so w.h.p. the number of cycles is $\sim \log n$ – *Key Observation.*
- Karp showed that the cost of patching is $o(1)$ w.h.p.

To patch vertex disjoint oriented cycles C_1, C_2 we choose edges $e_i = (x_i, y_i) \in C_i, i = 1, 2$. We delete e_1, e_2 from C_1, C_2 and then add the edges $(x_1, y_2), (x_2, y_1)$ to create a single cycle covering the vertices in C_1 or C_2. In this way, Karp proved

Theorem 7.2 (Karp [76]). *W.h.p.* $GAP = T(C) - A(C) = o(1)$.

This was improved:

Theorem 7.3 (Karp and Steele [79]). *W.h.p.* $GAP = T(C) - A(C) = O(n^{-1/2})$.

By making the cycles large before doing the patching we have

Theorem 7.4 (Dyer and Frieze [47]). *W.h.p.* $GAP = T(C) - A(C) = o\left(\frac{\log^4 n}{n}\right)$.

With more care we get

Theorem 7.5 (Frieze and Sorkin [58]). *W.h.p.* $GAP = T(C) - A(C) = O\left(\frac{\log^2 n}{n}\right)$.

The main tool in the improvements to Karp and Steele comes from cheaply transforming the cycle cover so that each cycle has length at least $n_0 = n \log\log n/\log n$.

Having increased the cycle size to $n_0 = n \log\log n/\log n$ we patch the cycles together using short edges. Each patch will cost $O(\log n/n)$ and so the patching cost is $o(\log^2 n/n)$. The assumption here is that after making all the cycles of the permutation large, there are many edges of length $O(\log n/n)$ that can be used to patch cycles together. Furthermore, the distribution of these edges is sufficiently nice so that we can claim: The probability we cannot patch a pair of cycles is at most

$$\left(1 - \Omega\left(\frac{\log^2 n}{n^2}\right)\right)^{\Omega(n_0^2)} = e^{-\Omega(\log^2 \log n)} = o(1/\log n).$$

In the same paper, Frieze and Sorkin observed that

Theorem 7.6. *W.h.p. the TSP can be solved exactly in* $2^{O(n^{1/2}\log^{O(1)} n)}$ *time.*

Having solved the assignment relaxation as a linear program, we search for a set of non-basic variables to increase from zero to one. We then argue that the distribution of the reduced costs are independent and uniform in $[o(1), 1]$.

Let

$$I_k = \frac{\log^2 n}{n}[2^{-k}, 2^{-k+1}].$$

Chernoff bounds for the binomial distribution imply that w.h.p. there are $\leq c_1 2^{-(k-1)} n \log n$ non-basic variables with reduced cost in $I_k, 1 \leq k \leq k_0 = \frac{1}{2}\log_2 n$ and $\leq 2c_1\sqrt{n}\log n$ non-basic variables with reduced cost $\leq c_1 \frac{(\log n)^2}{n^{3/2}}$.

Thus w.h.p. we need only check the following number of sets:

$$2^{2c_1\sqrt{n}\log n} \prod_{k=1}^{k_0} \sum_{t=1}^{2^k} \binom{c_1 2^{-(k-1)} n \log n}{t} = e^{O(\sqrt{n}\log^{O(1)} n)}$$

8. Random *k*-SAT

This is the name given to random instances of a version of the Satisfiabilty problem for Boolean formulae in conjunctive normal form. More precisely, we have a set of *variables* $V = \{x_1, x_2, \ldots, x_n\}$. The associated set of *literals* is $L = \{x_1, \bar{x}_1, x_2, \bar{x}_2, \ldots, x_n, \bar{x}_n\}$. A *clause* is a subset of L.

An instance I of k-SAT is defined as follows: Clauses $C_1, C_2, \ldots, C_m$ where $C_i \subseteq L$, $|C_i| = k$, $i = 1, 2, \ldots, m$.

A *truth assignment* is a map $\phi : L \to \{0, 1\}$ such that $\phi(\bar{x}_j) = 1 - \phi(x_j)$ for $j = 1, 2, \ldots, n$.

ϕ satisfies I if $1 \in \phi(C_i)$ for $i = 1, 2, \ldots, m$.

The k-SAT problem: Determine whether or not there is a satisfying assignment for I. It is solvable in polynomial time for $k \leq 2$. It is NP-hard for $k \geq 3$.

For a random instance I, we choose literals $\ell_1, \ell_2, \ldots, \ell_k$ independently and uniformly for each C_i, without replacement.

8.1. Bounds on the threshold. Let $m = cn$. Then for a fixed ϕ,

$$\mathbf{Pr}(\phi \text{ satisfies } I) = \left(1 - \frac{1}{2^k}\right)^m .$$

So, if Z is the number of satisfying assignments,

$$\mathbf{Pr}(\exists\phi \text{ satisfying } I) \leq \mathbf{E}(Z) = 2^n \left(1 - \frac{1}{2^k}\right)^m = \left(2\left(1 - \frac{1}{2^k}\right)^c\right)^n .$$

So I is unsatisfiable w.h.p. if $c > 2^k \log 2$. The hard question now is what is the smallest value of c for which I is unsatisfiable w.h.p. It is natural to make the following conjecture.

Conjecture: $\exists c_k$ such that if $m = cn$ then

$$\lim_{n\to\infty} \mathbf{Pr}(I \text{ is satisfiable}) = \begin{cases} 1 & c < c_k \\ 0 & c > c_k \end{cases}$$

Friedgut [53] has come close to proving this, in the same sense that he came close to proving the existence of a sharp threshold for graph coloring.

The conjecture is true for $k = 2$. It is known that $c_2 = 1$. Now if $m = cn$ and

$$\mathbf{Pr}(Z > 0) \geq \frac{\mathbf{E}(Z)^2}{\mathbf{E}(Z^2)} \tag{8.1}$$

and if the RHS here is bounded below then Friedgut's result will imply that $c \leq c_k$.

However, with Z equal to the number of satisfying assignments, the second moment method fails in the sense that the RHS of (8.1) tends to zero. Achlioptas and Peres [3] replace Z by

$$Z_1 = \sum_{\phi \text{ satisfies } I} \gamma^{H(\phi,I)}$$

where $H(\phi, I) =$ # true literals - # false literals in I for ϕ.

Using a careful choice of $0 < \gamma < 1$ they proved

Theorem 8.1. *If*

$$c < 2^k \log 2 - (k+1)\frac{\log 2}{2} - 1 - o_k(1)$$

then I is satisfiable w.h.p.

Using a more complicated random variable and doing more conditioning, but still using the second moment method, Coja-Oghlan and Panagiotou [39] proved that if

Theorem 8.2.

$$c < 2^k \log 2 - 3\frac{\log 2}{2} - 1 - o_k(1)$$

then I is satisfiable w.h.p.

Very recently, Coja-Oghlan [34] tightened this to

Theorem 8.3.

$$c < 2^k \log 2 - \frac{1 + \log 2}{2} - o_k(1)$$

then I is satisfiable w.h.p.

Finding c_k for $k = O(1)$ is a major open problem. If we allow k to grow then things become simple: Coja-Oghlan and Frieze [37] proved

Theorem 8.4. *Suppose that $k - \log_2 n \to \infty$ and that $m = 2^k(n \ln 2 + c)$ for an absolute constant c. Then,*

$$\lim_{n\to\infty} \mathbf{Pr}(I_m \textit{ is satisfiable}) = 1 - e^{-e^{-c}}$$

8.2. Algorithms.

We first consider *Greedy Algorithms*:

These start with no values assigned to the variables.

Then, they repeatedly, choose a random clause C and assign a value to a variable of C to satisfy it.

The number of variables in the problem goes down by one.

Some clauses get satisfied and disappear from the problem, others shrink in size by one.

Caveat: If there are "small" clauses be careful. For example if there is a clause of size one, then one is forced to assign a particular value to the variable it contains. Of course, this might create an empty clause, if the current set of clauses contains a pair $\{x_j\}, \{\bar{x}_j\}$.

Repeat until all of the clauses are satisfied (**success**) or there is a clause remaining that is empty (**fail**).

Most of these greedy algorithms find a satisfying assignment w.h.p. provided there are at most $\frac{c2^k}{k}n$ clauses, for small enough c.

A notable exception is the algorithm of Coja-Oghlan [33] which finds a satisfying assignment w.h.p. provided there are at most $\frac{(1-\epsilon)2^k \log k}{k}n$ clauses.

We now consider *Walksat*

Start with the "all true" assignment: $\phi(x_j) = 1$ for $j = 1, 2, \dots, n$

Repeat

Choose an unsatisfied clause C

Choose a random variable from C and change its assigned value

Until instance is satisfied.

If the instance is satisfiable, then Walksat will *eventually* find a solution.

Theorem 8.5 (Papadimitriou [96]). *For* arbitrary *cases of 2-SAT, the expected time to finish is polynomial.*

For random instances of 3-SAT:

Theorem 8.6 (Aleknovich and Ben-Sasson [7]). *Walksat solves a random instance of 3-SAT in polynomial time w.h.p. for $m < 1.67n$.*

The argument in [7] does not give a very good result for large k.

Theorem 8.7 (Coja-Oghlan, Feige, Frieze, Krivelevich and Vilenchik [36]). *For large k, Walksat solves a random instance of k-SAT in polynomial time w.h.p. for $m/n \le c2^k/k^2$.*

This was subsequently improved to

Theorem 8.8 (Coja-Oghlan and Frieze [38]). *For large k, Walksat solves a random instance of k-SAT w.h.p. for $m/n \le c2^k/k$.*

Outline proof of Theorem 8.7.

$$A_0 = \bigcup_{C \subseteq \bar{V}} C; \quad A_i = \left\{ x : \exists C \ni \bar{x} \text{ and } C \cap V \subseteq \bigcup_{j \leq i-1} A_j \right\}; \quad A = \bigcup_{i \geq 0} A_i.$$

Fact 1 Walksat only changes the truth value of variables in A.

Fact 2 So if $C \setminus A \neq \emptyset$, then C remains satisfied throughout.

Fact 3 $\forall C \subseteq A \, \exists L_C \subseteq C, |L_C| = 2k/3$ such that $C \neq C'$ implies $L_C \cap L_{C'} = \emptyset$.

Now define assignment σ_A:

$$\sigma_A(x) = \begin{cases} 1 & x \notin A \text{ or } x \in \bigcup_{C \subseteq A} L_C \\ 0 & \bar{x} \in \bigcup_{C \subseteq A} L_C \\ 1 & Otherwise \end{cases}$$

σ_A is a satisfying assignment. Now consider the Hamming distance between the current assignment σ_W of Walksat and σ_A. An iteration of Walksat reduces this by one with probability at least 2/3 and so by properties of simple random walk, this distance becomes zero in $O(n)$ time w.h.p., (unless another satisfying assignment is found). This is similar to the idea used by Papadimitriou [96] for 2-SAT.

8.3. Unsatisfiable instances. When the number of clauses is greater than the (conjectured) satisfiability threshold then one is interested in the time taken to prove that such an instance is unsatisfiable. A seminal paper by Chvátal and Szemerédi [32] showed that if $c > 0.7 \times 2^k$ is constant then w.h.p. any resolution proof of unsatisfiability of k-SAT must be exponentially long. A nice exposition of this and related results is given in Ben-Sasson and Wigderson [15].

9. Conclusions

We have hopefully shown a glimpse of an interesting area of research at the intersection of Combinatorics, Probability and Computing. Because of space limitations, we have omitted many things that we might have discussed and that the reader might find worth pursuing. As examples we have

(i) Perfect matchings in random hypergraphs, Johannson, Kahn and Vu [70];

(ii) Graph Property Resilience e.g. Sudakov and Vu [110], Ben-Shimon, Krivelevich and Sudakov [16];

(iii) Ramsey properties of random graphs e.g. Rödl and Ruciński [103];

(iv) Extremal properties of random graphs e.g. Conlon and Gowers [41], Schacht [105], Balogh, Morris and Samotij [11] and Saxton and Thomason [104];

(v) Random subgraphs of arbitrary graphs of large degree e.g. Krivelevich, Lee and Sudakov [84], Riordan [100], Frieze and Krivelevich [57];

(vi) Smoothed Analysis of the Simplex Algorithm for Linear Programs, Spielman and Teng [108] or Vershynin [112];

(vii) Integer Feasibility of Random Polytopes, Chandrasekaran and Vempala [30];

(viii) Nash Equilibria in Random Matrix Games, Bárány, Vempala and Vetta [12];

(ix) Random Knapsack Problems, Lueker [92], Goldberg and Marchetti-Spaccemela [61], Beier and B. Vöcking [14];

(x) The random graph $G_{K,p} = ([n], E_p)$ where K is a down monotone convex body in the non-negative orthant of $\mathbf{R}^n$ and $E_p = \left\{e \in \binom{[n]}{2} : X_e \leq p\right\}$ where X is chosen uniformly at random from K, see Frieze, Vera and Vempala [60];

(xi) Models of real world networks e.g. Bollobás and Riordan [29], Cooper and Frieze [44];

(xii) Random Simplicial Complexes e.g. Linial and Meshulam [88] or Kahle [72];

(xiii) Achlioptas processes e.g. Bohman and Frieze [18] or Kang, Perkins and Spencer [73];

(xiv) Random groups e.g. Gromov [65], Ollivier [95] or Antoniuk, Friedgut and Łuczak [8].

We conclude by mentioning some open questions.

P1 Find a polynomial time algorithm that w.h.p. finds a planted clique of size $o(n^{1/2})$ in $G_{n,1/2}$. (Negative results on the planted clique problem are given in [52]).

P2 Find the precise threshold for the k-colorability of the random graph $G_{n,p}$. Find a polynomial time algorithm that optimally colors $G_{n,p}$ w.h.p. or prove that this is impossible under some accepted complexity conjecture.

P3 Find the precise threshold for the satisfiability of random k-SAT. Find a polynomial time algorithm that determines the satisfiability of random k-SAT w.h.p. or prove that this is impossible under some accepted complexity conjecture.

P4 Prove that $\lim_{n\to\infty} \mathbf{Pr}(G_{n,cn;3} \text{ is Hamiltonian}) = 1$ for $c > 3/2$.

P5 Determine whether or not solving random asymmetric TSPs with independent costs by branch and bound runs in polynomial time w.h.p. when the bound used is the assignment problem value.

P6 Analyse the *ordinary* simplex algorithm on random instances.

P7 Let M be randomly chosen from the set of $n \times n$ symmetric $\{0, 1\}$ matrices with $r \geq 3$ ones in each row and column. Prove that M is non-singular w.h.p.

P8 Find a heuristic for the TSP in the unit square that w.h.p. comes with n^α of the optimum, where $0 < \alpha < 1/2$ is constant.

P9 Determine the constant β in Theoerem 7.1.

P10 Determine the asymptotics for the value of a random multi-dimensional assignment problem and find asymptotically optimal heuristics, see Frieze and Sorkin [59].

P11 Determine the threshold for a random subgraph of the n-cube to be Hamiltonian. See Bollobás [23] for the existence of a perfect matching.

Acknowledgements. The author was supported in part by NSF grants DMS0753472 and CCF1013110.

References

[1] D. Achlioptas and E. Friedgut, *A sharp threshold for k-colorability*, Random Structures and Algorithms **14** (1999), 63–70.

[2] D. Achlioptas and A. Naor, *The two possible values of the chromatic number of a random graph*, Annals of Mathematics **162** (2005), 1333–1349.

[3] D. Achlioptas and Y. Peres, *The threshold for random k-SAT is* $2^k \ln 2 - O(k)$, Journal of the AMS **17** (2004), 947–973.

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

[5] D. Aldous, *Asymptotics in the random assignment problem*, Probability Theory and Related Fields **93** (1992), 507–534.

[6] ______, *The $\zeta(2)$ limit in the random assignment problem*, Random Structures and Algorithms **18** (2001,) 381–418.

[7] M. Alekhnovich and E. Ben-Sasson, *Linear upper bounds for random walk on small density random 3-cnfs*, SIAM Journal on Computing **36** (2007), 1248–1263.

[8] S. Antoniuk, E. Friedgut, and T. Łuczak, *A sharp threshold for collapse of the random triangular group*.

[9] D.L. Applegate, R.E. Bixby, V. Chvátal, and W.J. Cook, *The Traveling Salesman Problem: A Computational Study*, Princeton University Press 2006.

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

[11] J. Balogh, R. Morris, and W. Samotij, *Independent sets in hypergraphs*.

[12] I. Bárány, S. Vempala, and A. Vetta, *Nash Equilibria in Random Games*, Random Structures and Algorithms **31** (2007), 391–405.

[13] J. Beardwood, J. H. Halton, and J. M. Hammersley, *The shortest path through many points*, Proceedings of the Cambridge Philosophical Society **55** (1959), 299–327.

[14] R. Beier and B. Vöcking, *Random knapsack in expected polynomial time*, Journal of Computer and System Science **69** (2004), 306–329.

[15] E. Ben-Sasson and A. Wigderson, *Short proofs are narrow-resolution made simple*, Journal of the ACM **48** (2001), 149–169.

[16] S. Ben-Shimon, M. Krivelevich, and B. Sudakov, *Local resilience and Hamiltonicity Maker-Breaker games in random regular graph*, Combinatorics, Probability and Computing **20** (2011), 173–211.

[17] A. Beveridge, A.M. Frieze, and C. McDiarmid, *Random minimum length spanning trees in regular graphs*, Combinatorica **18**, 311–333.

[18] T. Bohman and A.M. Frieze, *Avoiding a giant component*, Random Structures and Algorithms **19** (2001), 75–85.

[19] ______, *Hamilton cycles in 3-out*, Random Structures and Algorithms **35** (2010), 393–417.

[20] B. Bollobás, *The evolution of sparse graphs*, In Graph Theory and Combinatorics,

Proceedings of a Cambridge Combinatorial Conference in honour of Paul Erdős (Bollobás, B., Ed.). Academic Press (1984), 35–57.

[21] ———, *Combinatorics: Set Systems, Hypergraphs, Families of Vectors and Combinatorial Probability*, Cambridge University Press, 1986.

[22] ———, *The chromatic number of random graphs*, Combinatorica **8** (1988), 49–56.

[23] ———, *Complete matchings in random subgraphs of the cube*, Random Structures and Algorithms **1** (1990), 95–104.

[24] B. Bollobás, C. Cooper, T.I. Fenner, and A.M. Frieze, *On Hamilton cycles in sparse random graphs with minimum degree at least k*, Journal of Graph Theory **34** (2000), 42–59.

[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] ———, *Hamilton cycles in random graphs with minimal degree at least k*, in A tribute to Paul Erdős, edited by A. Baker, B .Bollobas, and A. Hajnal (1990), 59–96.

[28] B. Bollobás and A.M. Frieze, *On matchings and hamiltonian cycles in random graphs*, Annals of Discrete Mathematics **28** (1985), 23–46.

[29] B. Bollobás and O. Riordan, *Mathematical results on scale-free random grpahs*, in Handbook of graphs and networks: from genome to the internet, S. Bornholdt and H.G. Schuster eds. (2002), 1–34.

[30] K. Chandrasekaran and S.Vempala, *Integer Feasibility of Random Polytopes*.

[31] P. Chebolu, A.M. Frieze, and P. Melsted, *Finding a Maximum Matching in a Sparse Random Graph in $O(n)$ Expected Time*, Journal of the ACM (2010), 161–172.

[32] V. Chvátal and E. Szemerédi, *Many Hard Examples for Resolution*, Journal of the ACM **35** (1988), 759–768.

[33] A. Coja-Oghlan, *A better algorithm for random k-SAT*, Proceedings of the 36th ICALP Conference (2009), 292–303.

[34] ———, *The asymptotic k-SAT threshold.*

[35] ———, *Upper-bounding the k-colorability threshold by counting covers*, Electronic Journal of Combinatorics 20:P32 (2013).

[36] A. Coja-Oghlan, U. Feige, A.M. Frieze, M. Krivelevich, and D. Vilenchik, *On smoothed k-CNF formulas and the Walksat algorithm*, Proceedings of the 20th ACM-SIAM Conference on Discrete AAlgorithms (2009), 451–460.

[37] A. Coja-Oghlan and A.M. Frieze, *Random k-SAT: the limiting probability for satisfiability for moderately growing k*, Electronic Journal of Combinatorics 15:N2 (2008).

[38] ———, *Analyzing Walksat on random formulas*, Proceedings of the 9th ANALCO (2012), 48–55.

[39] A. Coja-Oghlan and K. Panagiotou, *Going after the k-SAT threshold*, Proceedings of the 45th ACM Symposium on the Theory of Computing (2013), 705–714.

[40] A. Coja-Oghlan and D. Vilenchik, *Chasing the k-colorability threshold*, Procedings of the 54th IEEE Symposium on the Foundations of Computing (2013), 380–389.

[41] D. Conlon and T. Gowers, *Combinatorial theorems in sparse random sets*.

[42] S. Cook, *The complexity of theorem proving procedures*, Proceedings of the Third Annual ACM Symposiumm on Theory of Computing (1971), 151–158.

[43] C. Cooper and A.M. Frieze, *On the number of hamilton cycles in a random graph*, Journal of Graph Theory **13** (1989), 719–735.

[44] ______, *On a general model of web graphs*, Random Structures and Algorithms **22**, (2003), 311–335.

[45] C. Cooper, A.M. Frieze, N. Ince, S. Janson, and J. Spencer, *On the length of a random minimum spanning tree*.

[46] C. Cooper, A.M. Frieze, and B. Reed, *Random regular graphs of non-constant degree: connectivity and Hamilton cycles*, Combinatorics, Probability and Computing **11**, 249–262.

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

[48] J. Edmonds, *Paths, trees, and flowers*, Canadian Journal of Mathematics **17** (1965), 125–130.

[49] P. Erdős and A. Rényi, *On random graphs I*, Publ. Math. Debrecen **6** (1959), 290–297.

[50] ______, *On the evolution of random graphs*, *Publ. Math. Inst. Hungar. Acad. Sci.* **5** (1960), 17–61.

[51] ______, *On the existence of a factor of degree one of a connected random graph*, Acta. Math. Acad. Sci. Hungar. **17** (1966), 359–368.

[52] V. Feldman, E. Grigorescu, L. Reyzin, and Y. Xiao, *Statistical Algorithms and a lower bound for detecting planted cliques*.

[53] E. Friedgut, *Sharp thresholds for graph properties, and the k-sat problem, with an appendix by Jean Bourgain*, Journal of the American Mathematical Society **12** (1999), 1017–1054.

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

[55] ______, *On a Greedy 2-Matching Algorithm and Hamilton Cycles in Random Graphs with Minimum Degree at Least Three*.

[56] A.M. Frieze and S. Haber, *An almost linear time algorithm for finding Hamilton cycles in sparse random graphs with minimum degree at least three*.

[57] A. Frieze and M. Krivelevich, *On the non-planarity of a random subgraph*, Combinatorics, Probability and Computing **22** (2013), 722–732.

[58] A.M. Frieze and G. Sorkin, *The probabilistic relationship between the assignment and asymmetric traveling salesman problems*, SIAM Journal on Computing **36** (2007), 1435–1452.

[59] ______, *Efficient algorithms for three-dimensional axial and planar random assignment problems*.

[60] A.M. Frieze, S. Vempala, and J. Vera, *Logconcave Random Graphs*, Electronic Journal of Combinatorics (2010).

[61] A.V. Goldberg and A. Marchetti-Spaccemela, *On finding the exact solution of a 0,1*

knapsack problem, Proceedings of the 16th Annual ACM Symposium on the Theory of Computing (1984), 359–368.

[62] R. Glebov and M. Krivelevich, *On the number of Hamilton cycles in sparse random graphs*, SIAM Journal on Discrete Mathematics **27** (2013), 27–42.

[63] R. Gomory, *Outline of an algorithm for integer solutions to linear programs*, Bulletin of the Amwerican Mathematical Society **64** (1958), 275–278.

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

[65] M. Gromov, *Asymptotic invariants of infinite groups. Geometric Group Theory*, London Mathematical Society Lecture Notes Series **182** (1993), 1–295.

[66] M. Held and R.M Karp, *A Dynamic Programming Approach to Sequencing Problems*, SIAM Journal of Applied Mathematics **10** (1962), 196–210.

[67] S. Janson, *The minimal spanning tree in a complete graph and a functional limit theorem for trees in a random graph, Random Structures Algorithms* **7** (1995), 337–355.

[68] ______, *One, two and three times* $\log n/n$ *for paths in a complete graph with random weights*, Combinatorics Probability and Computing **8** (1999), 347–361.

[69] S. Janson, D.E. Knuth, T. Łuczak and B. Pittel, *The Birth of the Giant Component*, Random Structures and Algorithms **4** (1993), 233–358.

[70] A. Johansson, J. Kahn, and V. Vu, *Factors in Random Graphs*, Random Structures and Algorithms **33** (2008), 1–28.

[71] S. Jukna, *Extremal Combinatorics: With Applications in Computer Science*, Springer 2011.

[72] M. Kahle, *Topology of random simplicial complexes: a survey.*

[73] M. Kang, W. Perkins, and J. Spencer, *The Bohman-Frieze process near criticality*, Random Structures and Algorithms **43** (2013), 221–250.

[74] R.M. Karp, *Reducibility Among Combinatorial Problems*, In Complexity of Computer Computations, R.E. Miller and J.W. Thatcher Eds. (1972), 85–103.

[75] ______, *Probabilistic Analysis of Partitioning Algorithms for the Traveling-Salesman Problem in the Plane*, Mathematics of Operations Research **2** (1977), 209–244.

[76] ______, *A patching algorithm for the non-symmetric traveling salesman problem*, SIAM Journal on Computing **8** (1979), 561–573.

[77] ______, *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.) Academic Press (1987), 1–4.

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

[79] 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., (1985), 181–205.

[80] L.G. Khacian, *A polynomial algorithm in Linear Programming*, Soviet Mathematics

Doklaidy **20** (1979,) 191–194.

[81] J. Kim and V. Vu, *Concentration of multi-variate polynomials and its applications*, Combinatorica **20** (2000,) 417–434.

[82] F. Knox, D. Kühn, and D. Osthus, *Edge-disjoint Hamilton cycles in random graphs, to appear in Random Structures and Algorithms*.

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

[84] M. Krivelevich, C. Lee, and B. Sudakov, *Long paths and cycles in random subgraphs of graphs with large minimum degree*.

[85] M. Krivelevich and W. Samotij, *Optimal packings of Hamilton cycles in sparse random graphs*, SIAM Journal on Discrete Mathematics **26** (2012), 964–982.

[86] M. Krivelevich, B. Sudakov, V. H. Vu and N. Wormald, *Random regular graphs of high degree*, Random Structures and Algorithms **18** (2001), 346–363.

[87] L. Levin, *Universal search problems*, Problems of Information Transmission **9** (1973), 115–116.

[88] N. Linial and R. Meshulam, *Homological connectivity of random 2-dimensional complexes*, Combinatorica **26** (2006), 475–487.

[89] S. Linusson and J. Wästlund, *A proof of Parisi's conjecture on the random assignment problem*, Probability Theory and Related Fields (2004), 419–440.

[90] L. Lovász, *Combinatorial problems and exercises*, Akadémiai Kiadó - North Holland, Budapest, 1979.

[91] T. Łuczak, *A note on the sharp concentration of the chromatic number of random graphs*, Combinatorica **11** (1991), 295–297.

[92] G.S. Lueker, *On the average distance between the solutions to linear and integer knapsack problems*, Applied Probability - Computer Science, The Interface **1** (1982), 489–504.

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

[94] C. Nair, B. Prabhakar, and M. Sharma, *Proofs of the Parisi and Coppersmith-Sorkin random assignment conjectures*, Random Structures and Algorithms **27** (2005), 413–44.

[95] Y. Ollivier, *Sharp phasr transition theorems for hyperbolicity of random groups*, Geometry and Functional Analysis **14** (2004), 595–679.

[96] C.H. Papadimitriou, *On selecting a satisfying truth assignment*, Proceeding sof the 32nd IEEE Symposium on the Foundations of Computing (1991), 163–169.

[97] G. Parisi, *A Conjecture on Random Bipartite Matching*, Physics e-Print archive (1998). http://xxx.lanl.gov/ps/cond-mat/9801176

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

[99] W. Rhee and M. Talagrand, *Martingale Inequalities and NP-Complete Problems*, Mathematics of Operations Research **12** (1987), 177–181.

[100] O. Riordan, *Long cycles in random subgraphs of graphs with large minimum degree*.

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

[102] ______, *Almost all regular graphs are Hamiltonian*, Random Structures and Algorithms **5** (1994), 363–374.

[103] V. Rödl and A. Ruciński, *Threshold functions for Ramsey properties*, Journal of the American Mathematical Society **8** (1995), 917–942.

[104] D. Saxton and A. Thomason, *Hypergraph containers*.

[105] M. Schacht, *Extremal results for random discrete structures*.

[106] R. Sedgewick and P. Flajolet, *An Introduction to the Analysis of Algorithms*, Pearson Education, 2013.

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

[108] D. Spielman and S-H. Teng, *Smoothed Analysis of Algorithms: Why The Simplex Algorithm Usually Takes Polynomial Time*, Journal of the ACM **51** (2004), 38–463.

[109] R. Stanley, *Enumerative Combinatorics, Volumes 1 and 2*, Cambridge University Press, 1999.

[110] B. Sudakov and V. Vu, *Local resilience of graphs*, Random Structures and Algorithms **33** (2008), 409–433.

[111] M. Talagrand, *Concentration of measure and isoperimetric inequalities in product spaces*, Publications Mathématiques de l'I.H.E.S. **81** (1995), 73–205.

[112] R. Vershynin, *Beyond Hirsch Conjecture: walks on random polytopes and smoothed complexity of the simplex method*, Procedings of the 47th Annual Symposium on Foundations of Computer Science (2006), 133–142.

[113] D.W. Walkup, *On the expected value of a random asignment problem*, SIAM Journal on Computing **8** (1979), 440–442.

[114] J. Wästlund, *An easy proof of the $\zeta(2)$ limit in the random assignment problem*, Electronic Communications in Probability (2009), 261–269.

Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh PA15213, USA.

E-mail: alan@random.math.cmu.edu