Random Structures and Algorithms

Alan Frieze

Department of Mathematical Sciences,

Carnegie Mellon University,

Pittsburgh, USA.

Contents of talk

(a) Random Discrete Structures

(b) Random Instances of the TSP in the unit square $[0,1]^2$

(c) The Random Graphs $G_{n,m}$ and $G_{n,p}$.

  (1) Evolution
  (2) Chromatic number
  (3) Matchings
  (4) Hamilton cycles

(d) Randomly edge weighted graphs

  1. Minimum Spanning Tree
  2. Shortest Paths
  3. 3-Dimensional Assignment Problem
  4. Random Instances of the TSP with independent costs

(e) Random k-SAT

(f) Open Problems

Random Discrete Structures

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

Random Discrete Structures

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 or a set of permutations) we have certain natural questions:

1. How big is Ω: Enumerative Combinatorics.

Random Discrete Structures

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 or a set of permutations) we have certain natural questions:

1. How big is Ω: Enumerative Combinatorics.
2. How big is the greatest element of Ω: Extremal Combinatorics.

Random Discrete Structures

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 or a set of permutations) 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.

Random Discrete Structures

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 or a set of permutations) 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. Analysis of algorithms aims to find the complexity of computational problems associated with the above topics.

Random Discrete Structures

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 or a set of permutations) 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. Analysis of algorithms aims to find the complexity of computational problems associated with the above topics.

This talk will be about Probabilistic Combinatorics/Analysis of Algorithms (average case).

Random Discrete Structures

There is unfortunately no time to discuss the Probabilistic Method where one uses probabilistic arguments to prove the existence of certain mathematical entities.

Contents of talk

(a) Random Discrete Structures

(b) Random Instances of the TSP in the unit square $[0,1]^2$

(c) The Random Graphs $G_{n,m}$ and $G_{n,p}$.

   (1) Evolution
   (2) Chromatic number
   (3) Matchings
   (4) Hamilton cycles

(d) Randomly edge weighted graphs

   1. Minimum Spanning Tree
   2. Shortest Paths
   3. 3-Dimensional Assignment Problem
   4. Random Instances of the TSP with independent costs

(e) Random k-SAT

(f) Open Problems

TSP in the Unit Square

It is a feature of modern computation that many of the problems we would like to solve seem hard, in some well-defined sense e.g. NP-hard. As such any algorithm is likely to take a large amount of time on some problems.

TSP in the Unit Square

It is a feature of modern computation that many of the problems we would like to solve seem hard, in some well-defined sense e.g. NP-hard. As such any algorithm is likely to take a large amount of time on some problems.

This of course refers to the worst-case scenario for a particular problem.

TSP in the Unit Square

It is a feature of modern computation that many of the problems we would like to solve seem hard, in some well-defined sense e.g. NP-hard. As such any algorithm is likely to take a large amount of time on some problems.

This of course refers to the worst-case scenario for a particular problem.

In practise, typical problems are often easy to satisfactorily solve.

TSP in the Unit Square

It is a feature of modern computation that many of the problems we would like to solve seem hard, in some well-defined sense e.g. NP-hard. As such any algorithm is likely to take a large amount of time on some problems.

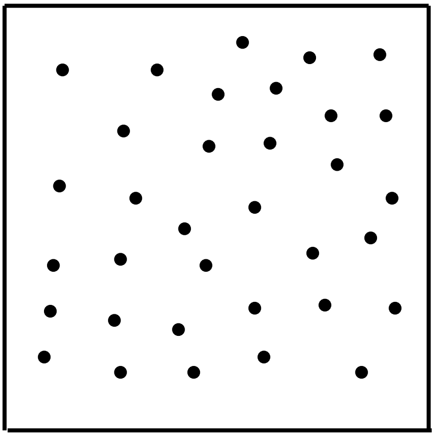
This of course refers to the worst-case scenario for a particular problem.

In practise, typical problems are often easy to satisfactorily solve.

Karp (1977) pioneered the idea of finding algorithms that work well on instances drawn from some natural probability distribution. He focussed first on the Travelling Salesperson Problem (TSP).

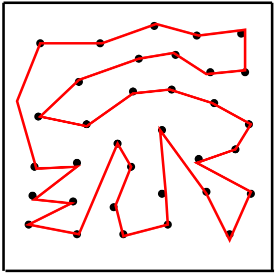
TSP in the Unit Square

Let $\mathcal{X} = X_1, X_2, \ldots, X_n$ be n points chosen independently and uniformly from $[0, 1]^2$.



TSP in the Unit Square

$X_1, X_2, \ldots, X_n$ are independently chosen, uniformly from $[0,1]^2$.



Let Z be the minimum total length of a closed path (tour) through $X_1, X_2, \ldots, X_n$.

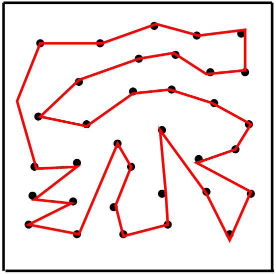
We consider the likely value of Z as $n \to \infty$.

Theorem (Beardwood, Halton and Hammersley (1959))

There exists an absolute constant $\beta > 0$ such

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

$X_1, X_2, \ldots, X_n$ are independently chosen, uniformly from $[0,1]^2$.

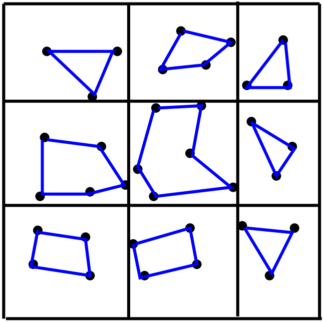


Let Z be the minimum total length of a closed path (tour) through $X_1, X_2, \ldots, X_n$.

We consider the likely value of Z as $n \to \infty$.

The precise value of β is unkown to this day.

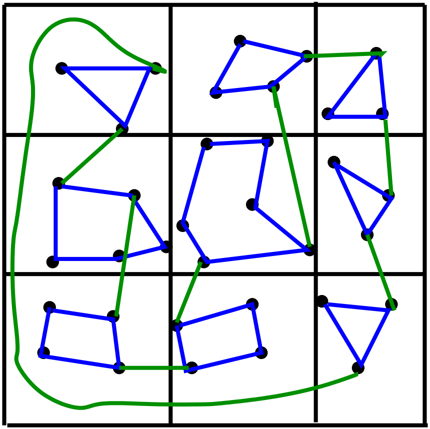
Karp (1977) described a heuristic that runs in polynomial time and w.h.p. produces a tour of length $Z + o(n^{1/2})$.



Sub-square size $(\log n/n)^{1/2}$

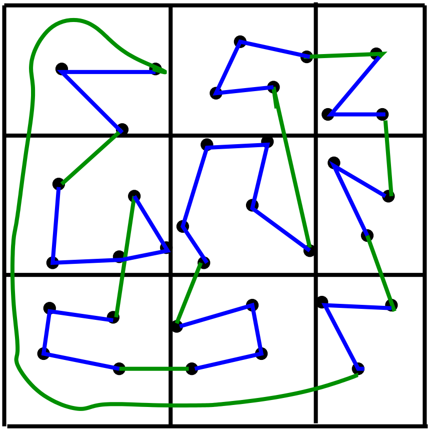
Solve the individual problems in each sub-square.

TSP in the Unit Square



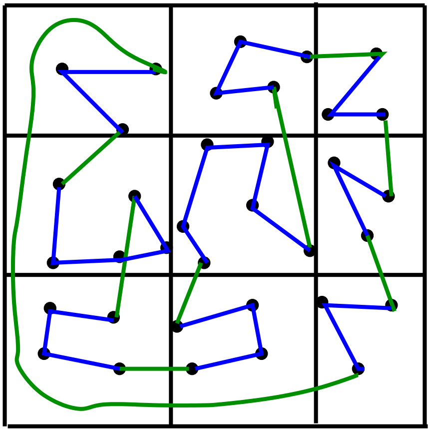
Connect up the smaller tours as shown, by green edges.

TSP in the Unit Square



Connect up the smaller tours as shown, by green edges.
Convert to tour by deleting excess edges.

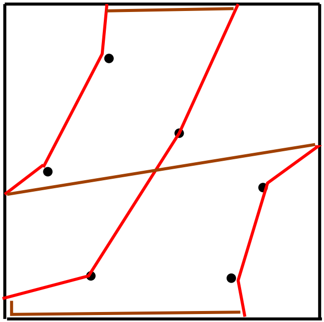
TSP in the Unit Square



The total length of the green edges is $O(n^{1/2}/L) = o(n^{1/2})$.

Travelling Salesman in the Unit Square:

Single Sub-Square



Red edges from optimal tour through all n points.
Red plus Brown edges at least as long as the one found in the sub-square by the algorithm.
Total length of brown edges is $O(n/L^2) \times Ln^{-1/2} = o(n^{1/2})$.

TSP in the Unit Square

Theorem (Karp (1977))

There is a polynomial time algorithm that w.h.p. finds a tour of length $(1+o(1))\beta n^{1/2}$.

Here w.h.p. (with high probability) means with probability $1-o(1)$ as $n \to \infty$.

TSP in the Unit Square

Theorem (Karp (1977))

There is a polynomial time algorithm that w.h.p. finds a tour of length $(1+o(1))\beta n^{1/2}$.

Here w.h.p. (with high probability) means with probability $1-o(1)$ as $n\to\infty$.

Note that Papadimitriou (1977) showed that the TSP restricted to Euclidean instances is still NP-hard.

Contents of talk

(a) Random Discrete Structures

(b) Random Instances of the TSP in the unit square $[0, 1]^2$

(c) The Random Graphs $G_{n,m}$ and $G_{n,p}$.

- (1) Evolution
- (2) Chromatic number
- (3) Matchings
- (4) Hamilton cycles

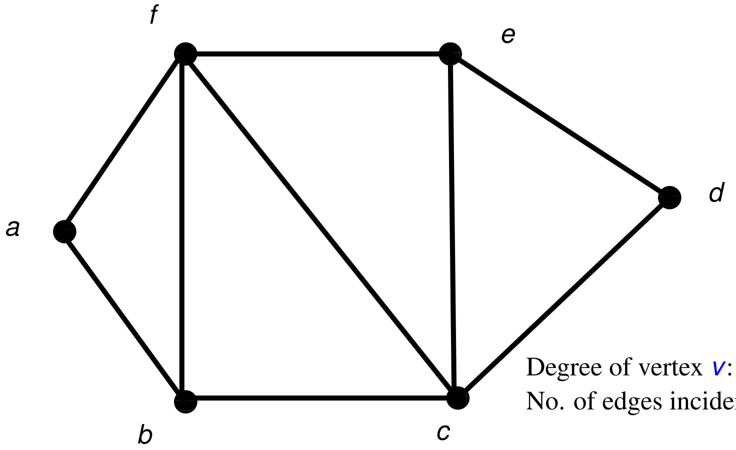
(d) Randomly edge weighted graphs

1. Minimum Spanning Tree
2. Shortest Paths
3. 3-Dimensional Assignment Problem
4. Random Instances of the TSP with independent costs

(e) Random k-SAT

(f) Open Problems

$G_{n,m}$ and $G_{n,p}$

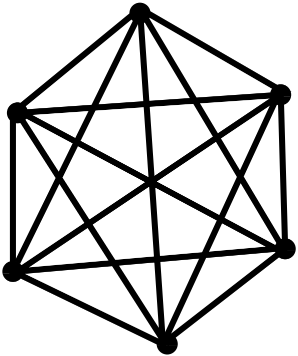


Degree of vertex v:
No. of edges incident with v

Graph $G = (V, E)$
Vertices $V = \{a, b, c, d, e, f\}$
Edges $E = \{\{a, b\}, \{a, f\}, \ldots, \{e, f\}\}$

$G_{n,m}$ and $G_{n,p}$

The complete graph K_n has vertex set $[n] = \{1, 2, \ldots, n\}$ and edge set $\binom{[n]}{2}$.



K_6

Choosing a graph at random

Choosing a graph at random

$G_{n,m}$: Vertex set $[n]$ and m random edges.

Choosing a graph at random

$G_{n,m}$: Vertex set $[n]$ and m random edges.

$G_{n,p}$: Each edge e of the complete graph K_n is included independently with probability $p = p(n)$.

Choosing a graph at random

$G_{n,m}$: Vertex set $[n]$ and m random edges.

$G_{n,p}$: Each edge e of the complete graph K_n is included independently with probability $p = p(n)$.

W.h.p. $G_{n,p}$ has $\sim \binom{n}{2}p$ edges, provided $\binom{n}{2}p \to \infty$

$p = 1/2$, each subgraph of K_n is equally likely.

If $m \sim \binom{n}{2}p$ then $G_{n,p}$ and $G_{n,m}$ have "similar" properties.

W.h.p. means with probability 1-o(1) as $n \to \infty$.

Contents of talk

(a) Random Discrete Structures

(b) Random Instances of the TSP in the unit square $[0, 1]^2$

(c) The Random Graphs $G_{n,m}$ and $G_{n,p}$.

  (1) Evolution
  (2) Chromatic number
  (3) Matchings
  (4) Hamilton cycles

(d) Randomly edge weighted graphs

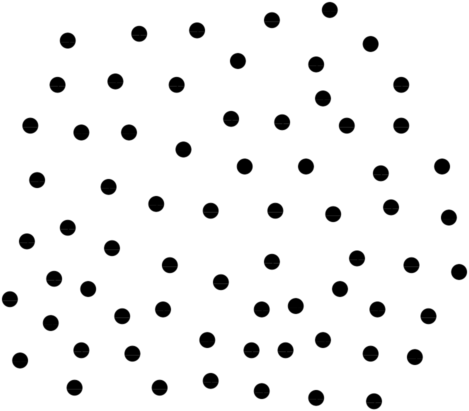
  1. Minimum Spanning Tree
  2. Shortest Paths
  3. 3-Dimensional Assignment Problem
  4. Random Instances of the TSP with independent costs

(e) Random k-SAT

(f) Open Problems

The evolution of a random graph

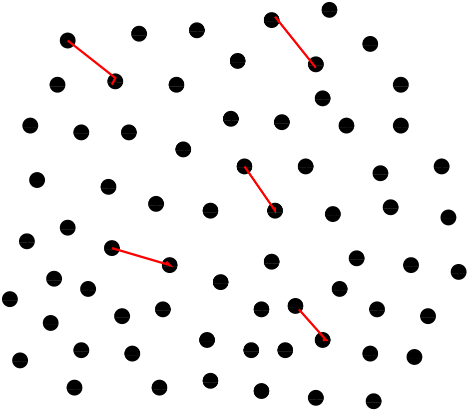
Graph process $G_0, G_1, \ldots$ where G_{i+1} is G_i plus a random edge.



In the beginning

The evolution of a random graph

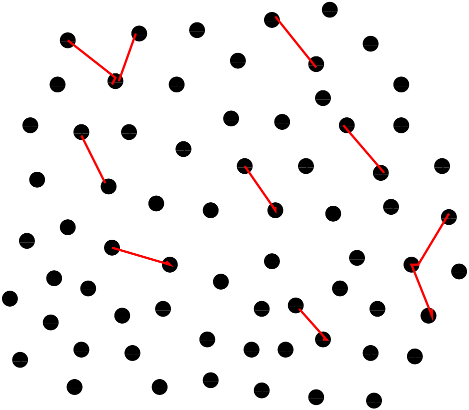
Graph process $G_0, G_1, \ldots$ where G_{i+1} is G_i plus a random edge.



$m = o(n^{1/2})$

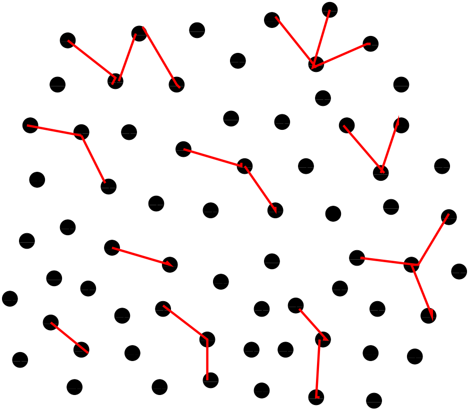
The evolution of a random graph

Graph process $G_0, G_1, \ldots$ where G_{i+1} is G_i plus a random edge.



$m = \omega(n^{1/2})$

Graph process $G_0, G_1, \ldots$ where G_{i+1} is G_i plus a random edge.



$m = \omega(n^{2/3})$

The evolution of a random graph

Erdős and Rényi (1960)

m	Structure of $G_{n,m}$ w.h.p.
$o(n^{1/2})$	Isolated edges and vertices

The evolution of a random graph

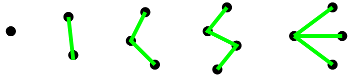
Erdős and Rényi (1960)

m	Structure of $G_{n,m}$ w.h.p.
$o(n^{1/2})$	Isolated edges and vertices
$\omega(n^{1/2})$	Isolated edges and vertices and paths of length 2

The evolution of a random graph

Erdős and Rényi (1960)

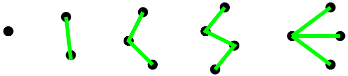
m	Structure of $G_{n,m}$ w.h.p.
$o(n^{1/2})$	Isolated edges and vertices
$\omega(n^{1/2})$	Isolated edges and vertices and paths of length 2
$\omega(n^{2/3})$	Components are of the form



The evolution of a random graph

Erdős and Rényi (1960)

m	Structure of $G_{n,m}$ w.h.p.
$o(n^{1/2})$	Isolated edges and vertices
$\omega(n^{1/2})$	Isolated edges and vertices and paths of length 2
$\omega(n^{2/3})$	Components are of the form



$\omega(n^{\frac{k-1}{k}})$	Components are trees with $1 \leq j \leq k+1$ vertices. Each possible such tree appears.

The evolution of a random graph

m	Structure of $G_{n,m}$ w.h.p.
$\frac{1}{2}cn$ $c < 1$	Mainly trees. Some unicyclic components. Maximum component size $O(\log n)$

The evolution of a random graph

m	Structure of $G_{n,m}$ w.h.p.
$\frac{1}{2}cn$ $c < 1$	Mainly trees. Some unicyclic components. Maximum component size $O(\log n)$
$\frac{1}{2}cn$ $c > 1$	Unique giant component of size $\sim \gamma(c)n$. Remainder almost all trees. Second largest component of size $O(\log n)$

The evolution of a random graph

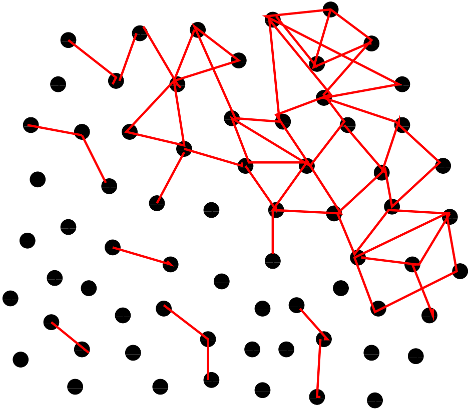
m	Structure of $G_{n,m}$ w.h.p.
$\frac{1}{2}cn$ $c < 1$	Mainly trees. Some unicyclic components. Maximum component size $O(\log n)$
$\frac{1}{2}cn$ $c > 1$	Unique giant component of size $\sim \gamma(c)n$. Remainder almost all trees. Second largest component of size $O(\log n)$
$\frac{1}{2}n$	Fascinating. Maximum component size order $n^{2/3}$. Has subsequently been the subject of more intensive study e.g. Janson, Knuth, Łuczak and Pittel (1993).

The evolution of a random graph

m	Structure of $G_{n,m}$ w.h.p.
$\frac{1}{2}cn$ $c > 1$	Unique giant component of size $\sim \gamma(c)n$. Remainder almost all trees. Second largest component 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 go extinct.

The evolution of a random graph



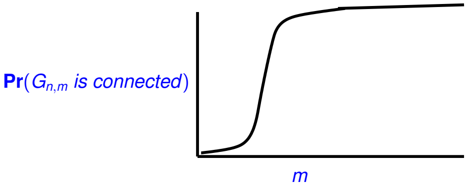
$m = cn/2,\ c > 1$

The evolution of a random graph

Theorem (Erdős and Rényi (1959))

$m = \frac{1}{2}n(\log n + c_n)$

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



Notice the sharp transition from disconnected to connected.

Connectivity threshold

$$p = (1+\epsilon)\frac{\log n}{n}, \quad \left(m = \frac{1+\epsilon}{2} n \log n.\right)$$

X_k = number of k-components, $1 \le k \le n/2$.
$X = X_1 + X_2 + \cdots + X_{n/2}$
$G_{n,p}$ is connected iff $X = 0$.

Connectivity threshold

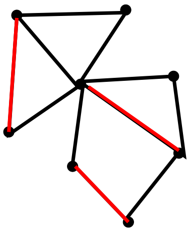
$$p = (1+\epsilon)\frac{\log n}{n}, \qquad \left(m = \frac{1+\epsilon}{2} n \log n.\right)$$

X_k = number of k-components, $1 \leq k \leq n/2$.
$X = X_1 + X_2 + \cdots + X_{n/2}$
$G_{n,p}$ is connected iff $X = 0$.

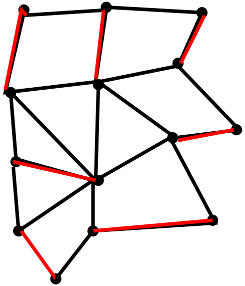
$$\begin{aligned}
\mathbf{Pr}(X \neq 0) &\leq \mathbf{E}(X) \\
&\leq \sum_{k=1}^{n/2} \binom{n}{k} k^{k-2} p^{k-1} (1-p)^{k(n-k)} \\
&\leq \frac{n}{\log n} \sum_{k=1}^{n/2} \left(\frac{e \log n}{n^{(1+\epsilon)(1-k/n)}} \right)^k \\
&\to 0.
\end{aligned}$$

Evolution: Matchings

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.



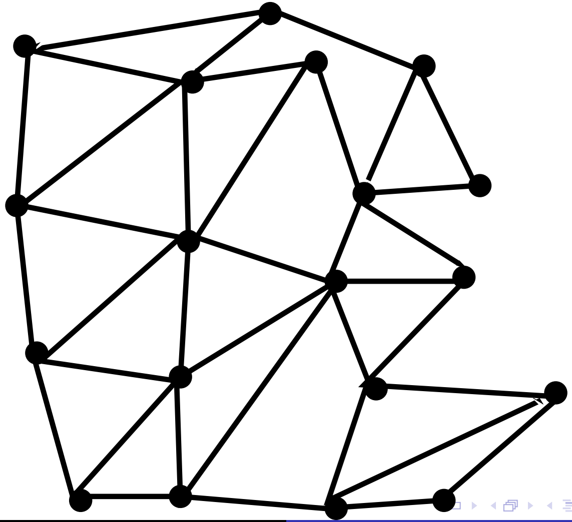
Matching



Perfect Matching

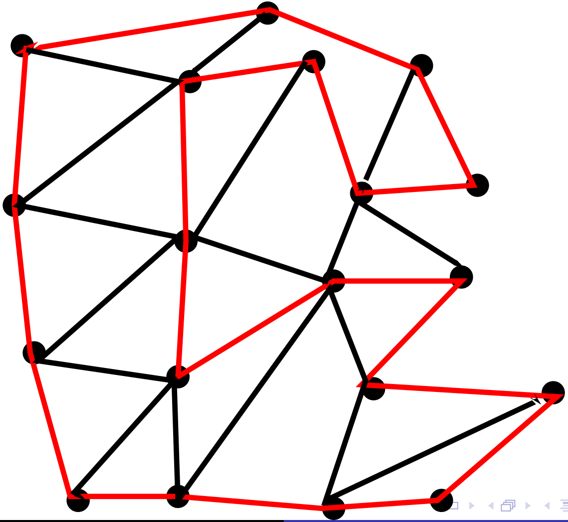
Evolution: Hamilton cycles

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



Evolution: Hamilton cycles

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



Evolution: Hitting Times

Consider $G_0, G_1, \ldots, G_m, \ldots,$: G_{i+1} is G_i plus a random edge. Let m_k denote the minimum m for which the minimum vertex degree $\delta(G_m) \geq k$.

Evolution: Hitting Times

Consider $G_0, G_1, \ldots, G_m, \ldots,$: G_{i+1} is G_i plus a random edge. Let m_k denote the minimum m for which the minimum vertex degree $\delta(G_m) \geq k$.

Theorem (Erdős and Rényi (1959))

W.h.p. m_1 is the "time" when G_m first becomes connected.

Evolution: Hitting Times

Consider $G_0, G_1, \ldots, G_m, \ldots$,: G_{i+1} is G_i plus a random edge. Let m_k denote the minimum m for which the minimum vertex degree $\delta(G_m) \geq k$.

Theorem (Erdős and Rényi (1959))

W.h.p. m_1 is the "time" when G_m first becomes connected.

Theorem (Erdős and Rényi (1966))

W.h.p. m_1 is the "time" when G_m first has a perfect matching.

Evolution: Hitting Times

Consider $G_0, G_1, \ldots, G_m, \ldots,$: G_{i+1} is G_i plus a random edge. Let m_k denote the minimum m for which the minimum vertex degree $\delta(G_m) \geq k$.

Theorem (Erdős and Rényi (1959))

W.h.p. m_1 *is the "time" when* G_m *first becomes connected.*

Theorem (Erdős and Rényi (1966))

W.h.p. m_1 *is the "time" when* G_m *first has a perfect matching.*

Theorem (Ajtai, Komlós, Szemerédi (1985), Bollobás (1984))

W.h.p. m_2 *is the "time" when* G_m *first has a Hamilton cycle.*

Evolution: Hitting Times

Consider $G_0, G_1, \ldots, G_m, \ldots,$: G_{i+1} is G_i plus a random edge. Let m_k denote the minimum m for which the minimum vertex degree $\delta(G_m) \geq k$.

Theorem (Cooper and Frieze (1989))

W.h.p. at "time" m_2, G_m has $(\log n)^{n-o(n)}$ Hamilton cycles.

Consider $G_0, G_1, \ldots, G_m, \ldots,$: G_{i+1} is G_i plus a random edge. Let m_k denote the minimum m for which the minimum vertex degree $\delta(G_m) \geq k$.

Theorem (Glebov and Krivelevich (2013))

W.h.p. at "time" m_2, G_m has $n!p^n e^{-o(n)}$ Hamilton cycles.

Evolution: Hitting Times

Consider $G_0, G_1, \ldots, G_m, \ldots$,: G_{i+1} is G_i plus a random edge. Let m_k denote the minimum m for which the minimum vertex degree $\delta(G_m) \geq k$.

Theorem (Bollobás and Frieze (1985))

W.h.p. at "time" $m_k, k = O(1)$, G_m has $\lfloor k/2 \rfloor$ disjoint Hamilton cycles plus a disjoint perfect matching if k is odd– Property $\mathcal{A}_k$.

Evolution: Hitting Times

Consider $G_0, G_1, \ldots, G_m, \ldots,$: G_{i+1} is G_i plus a random edge. Let m_k denote the minimum m for which the minimum vertex degree $\delta(G_m) \geq k$.

Theorem (Bollobás and Frieze (1985))

W.h.p. at "time" $m_k, k = O(1)$, G_m has $\lfloor k/2 \rfloor$ disjoint Hamilton cycles plus a disjoint perfect matching if k is odd– Property $\mathcal{A}_k$.

Theorem (Knox, Kühn and Osthus (2012))

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 (Krivelevich and Samotij (2012))

W.h.p. G_m has property $\mathcal{A}_\delta$ for $\frac{1}{2} n \log n \leq m \leq n^{1+\epsilon}$.

Contents of talk

(a) Random Discrete Structures

(b) Random Instances of the TSP in the unit square $[0,1]^2$

(c) The Random Graphs $G_{n,m}$ and $G_{n,p}$.

  (1) Evolution
  (2) Chromatic number
  (3) Matchings
  (4) Hamilton cycles

(d) Randomly edge weighted graphs

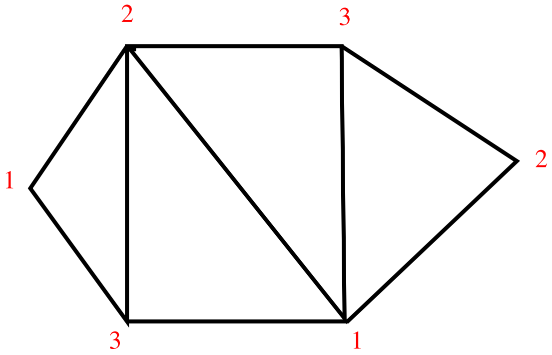
  1. Minimum Spanning Tree
  2. Shortest Paths
  3. 3-Dimensional Assignment Problem
  4. Random Instances of the TSP with independent costs

(e) Random k-SAT

(f) Open Problems

Chromatic Number

A proper k-coloring of a graph G is a map $f : V \rightarrow [k]$ 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.

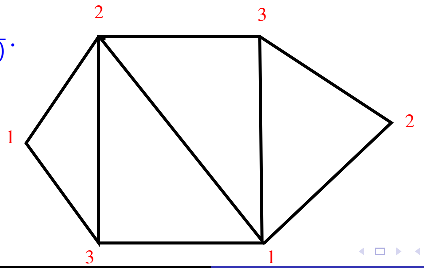
A set of vertices $S \subseteq V$ is independent if $v, w \in S$ implies that $\{v, w\}$ is not an edge.

In a proper k-coloring, each color class is an independent set.

The independence number $\alpha(G)$ is the size of the largest independent set.

Thus

$$\chi(G) \geq \frac{|V|}{\alpha(G)}.$$



Theorem (Matula (1970))

W.h.p. the maximum size $\alpha(G_{n,1/2})$ *of an independent set is*

$$2\log_2 n - 2\log_2\log_2 n + O(1).$$

Theorem (Matula (1970))

W.h.p. the maximum size $\alpha(G_{n,1/2})$ of an independent set is

$$2\log_2 n - 2\log_2\log_2 n + O(1).$$

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

Greedy Algorithm:
Start with $I = \{1\}$.
Repeatedly add $v \notin I$ that is not adjacent to I, until no such v can be found.

Chromatic Number

Theorem (Matula (1970))

W.h.p. the maximum size $\alpha(G_{n,1/2})$ of an independent set is

$$2\log_2 n - 2\log_2\log_2 n + O(1).$$

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

Greedy Algorithm:
Start with $I = \{1\}$.
Repeatedly add $v \notin I$ that is not adjacent to I, until no such v can be found.

After k successful steps, $\mathbf{E}(\#\text{ choices for } v) \sim n2^{-k}$.

Theorem (Matula (1970))

W.h.p. the maximum size $\alpha(G_{n,1/2})$ *of an independent set is*

$$2\log_2 n - 2\log_2\log_2 n + O(1).$$

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 $\epsilon > 0$.

Theorem (Matula (1970))

W.h.p. the maximum size $\alpha(G_{n,1/2})$ of an independent set is

$$2\log_2 n - 2\log_2\log_2 n + O(1).$$

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 $\epsilon > 0$.

It may not be possible to find such an independent set in polynomial time w.h.p.

Chromatic Number

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

Chromatic Number

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

Theorem (Bollobás and Erdős (1976), Grimmett and McDiarmid (1975))

W.h.p. a simple greedy algorithm uses $\sim \frac{n}{\log_2 n}$ *colors.*

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

Theorem (Bollobás and Erdős (1976), Grimmett and McDiarmid (1975))

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 $(1-\epsilon)n/\log_2 n$ colors in polynomial time.

Chromatic Number

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

Theorem (Bollobás and Erdős (1976), Grimmett and McDiarmid (1975))

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 $(1-\epsilon)n/\log_2 n$ colors in polynomial time.

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

Chromatic Number

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

Chromatic Number

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.

Chromatic Number

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

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

“Discovered” by Shamir and Spencer (1987), Rhee and Talagrand (1988), they have had a profound effect on our area.

Chromatic Number

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

$$\mathbf{Pr}(Z = 0) \leq e^{-2\mathbf{E}(Z)^2/N}.$$

"Discovered" by Shamir and Spencer (1987), Rhee and Talagrand (1988), they have had a profound effect on our area.

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

$$\mathbf{Pr}(Z = 0) \leq e^{-2\mathbf{E}(Z)^2/N}.$$

Further inequalities by Talagrand (1995) and Kim and Vu (2000) have been extremely useful.

Bollobás (1988) proved

Theorem

$\chi(G_{n,1/2}) \sim \frac{n}{2\log_2 n}$ *w.h.p.*

Chromatic Number

Bollobás (1988) proved

Theorem

$\chi(G_{n,1/2}) \sim \frac{n}{2\log_2 n}$ *w.h.p.*

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 $|S_i \cap S_j| \leq 1$.

Bollobás (1988) proved

Theorem

$\chi(G_{n,1/2}) \sim \frac{n}{2\log_2 n}$ *w.h.p.*

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 $|S_i \cap S_j| \leq 1$.

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

So,

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

Bollobás (1988) proved

Theorem

$\chi(G_{n,1/2}) \sim \frac{n}{2\log_2 n}$ *w.h.p.*

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 $|S_i \cap S_j| \leq 1$.

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

So,

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

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.

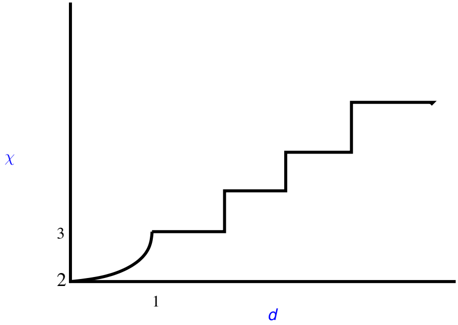
Chromatic Number

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.

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

Friedgut (1999), Achlioptas and Friedgut (1999) came close to proving this.



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

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

Friedgut (1999), Achlioptas and Friedgut (1999) came close to proving this.

Theorem (Łuczak (1991))

W.h.p. $\chi(G_{n,d/n})$ takes one of two values.

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

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

Friedgut (1999), Achlioptas and Friedgut (1999) came close to proving this.

Surprisingly, using a second moment method we get

Theorem (Achlioptas and Naor (2005))

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

Theorem (Achlioptas and Naor (2005))

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

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 $d < 2(k-1)\log(k-1)$.
Now use results on sharpness of threshold.

Theorem (Achlioptas and Naor (2005))

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

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 $d < 2(k-1)\log(k-1)$.
Now use results on sharpness of threshold.

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

Theorem (Achlioptas and Naor (2005))

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

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 $d < 2(k-1)\log(k-1)$.
Now use results on sharpness of threshold.

The result has been extended to hypergraphs:
Dyer, Frieze and Greenhill (2014).

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

Theorem (Achlioptas and Naor (2005))

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 (Coja-Oghlan and Vilenchik (2013))

(a) 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 asymptotic density one in* R_+.

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

Chromatic Number

Upper bound on d_k: Let $X_k(d)$ denote the number of k-colorings of $G_{n,d/n}$. Then

$$d > 2k \log k - \log k \text{ implies } \mathbf{E}(X_k(d)) \to 0$$

and therefore

$$d_k < 2k \log k - \log k.$$

Chromatic Number

Upper bound on d_k:

$$d_k < 2k \log k - \log k.$$

Theorem (Coja-Oghlan (2014))

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

Upper bound on d_k:

$$d_k < 2k \log k - \log k.$$

Theorem (Coja-Oghlan (2014))

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

This problem has attracted the attention of Statistical Physicists where colors are synonyms for spins. Coja-Oghlan's proof is motivated by physicists conjectures about the geometry of the set of k-colorings near the threshold. His upper bound matches a physics prediction.

Chromatic Number

Upper bound on d_k:

$$d_k < 2k\log k - \log k.$$

Theorem (Coja-Oghlan (2014))

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

For large k, the value of d_k is now known within an interval of size less than 0.39.

Contents of talk

(a) Random Discrete Structures

(b) Random Instances of the TSP in the unit square $[0,1]^2$

(c) The Random Graphs $G_{n,m}$ and $G_{n,p}$.

  (1) Evolution
  (2) Chromatic number
  (3) Matchings
  (4) Hamilton cycles

(d) Randomly edge weighted graphs

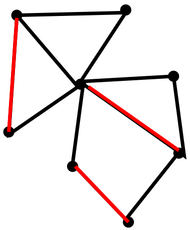
  1. Minimum Spanning Tree
  2. Shortest Paths
  3. 3-Dimensional Assignment Problem
  4. Random Instances of the TSP with independent costs

(e) Random k-SAT

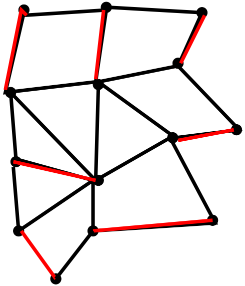
(f) Open Problems

Matchings

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.



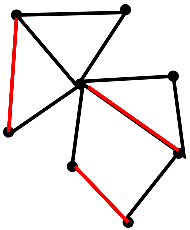
Matching



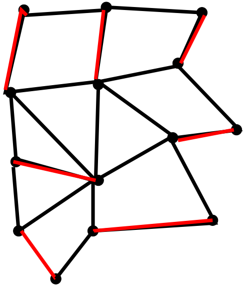
Perfect Matching

Matchings

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.



Matching



Perfect Matching

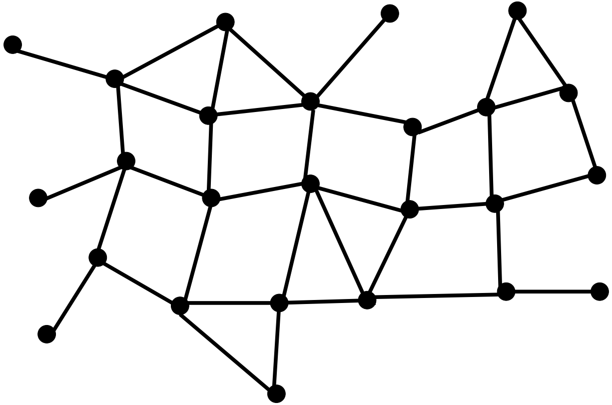
A largest matching can be found in polynomial time Edmonds (1965).

Karp and Sipser (1981) proposed the following greedy algorithm for finding a large matching:

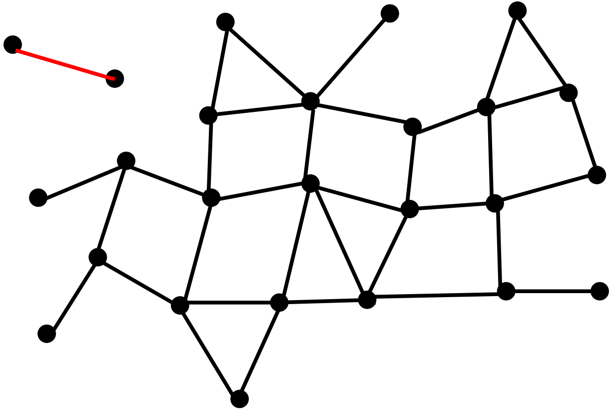
KSGREEDY

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;
A2: Otherwise, (no vertices of degree one) choose $e = \{x, y\} \in E$ randomly
$G \leftarrow G \setminus \{x, y\}$;
$M \leftarrow M \cup \{e\}$
end;
Output M
end

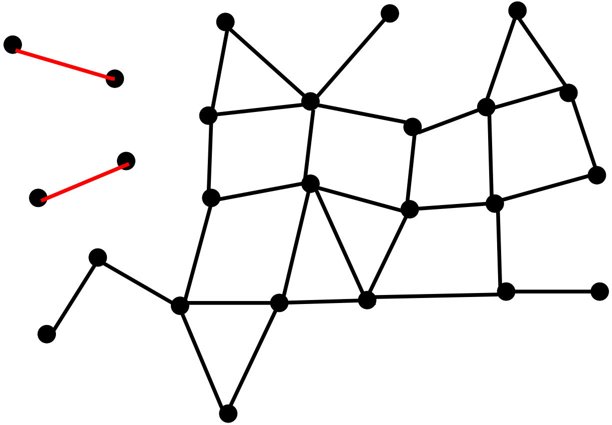
Matchings



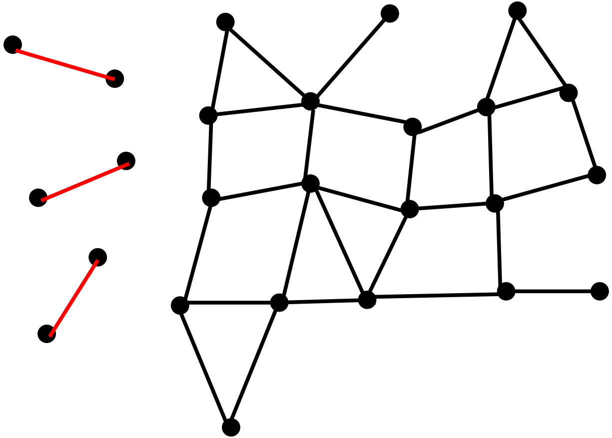
Matchings



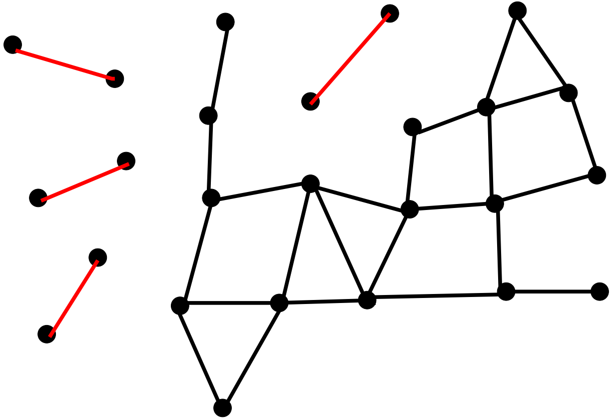
Matchings



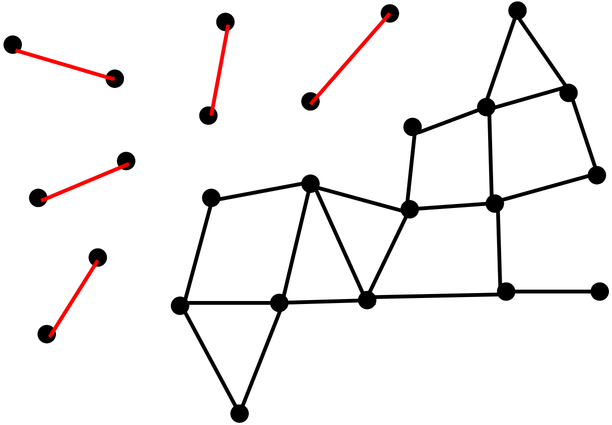
Matchings



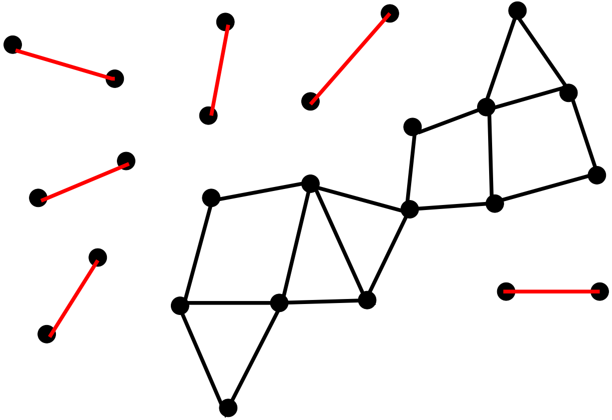
Matchings



Matchings

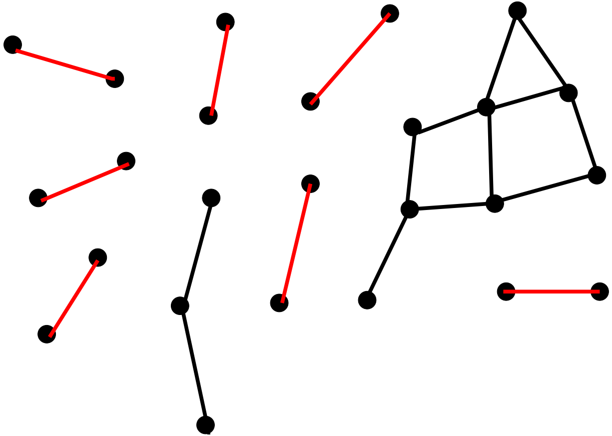


Matchings

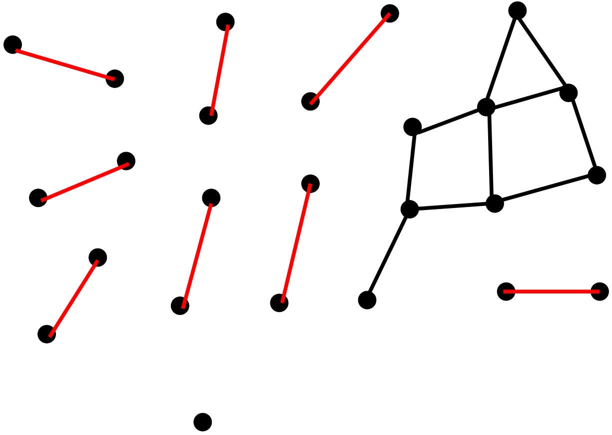


End of Phase 1

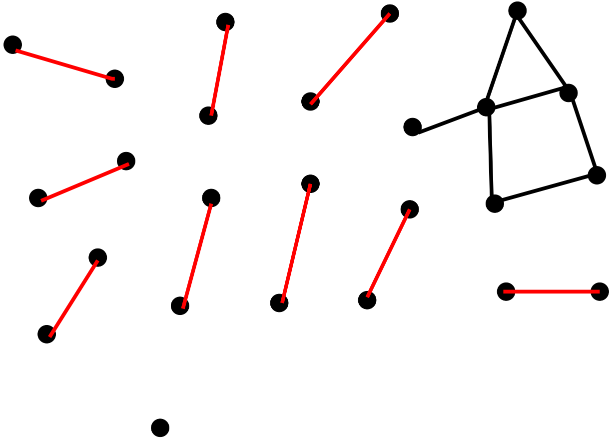
Matchings



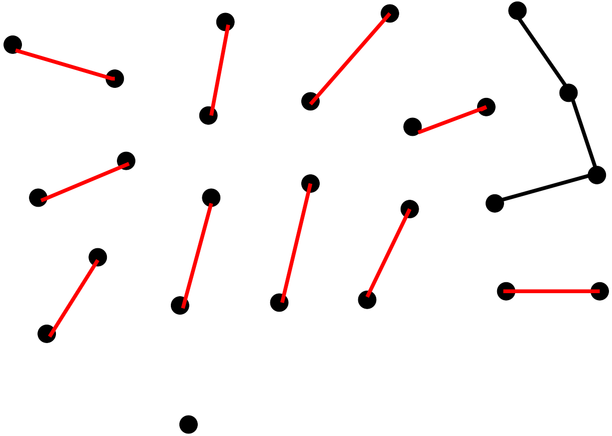
Matchings



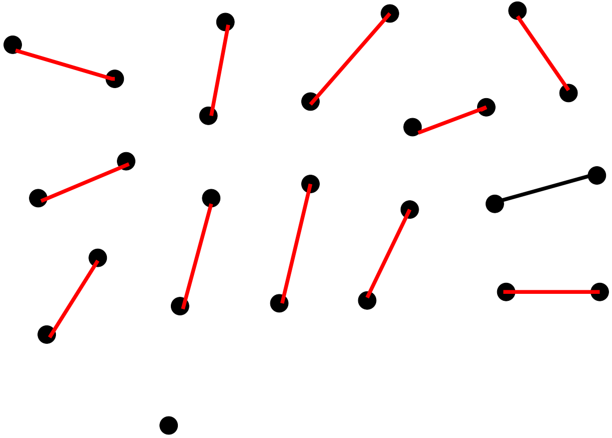
Matchings



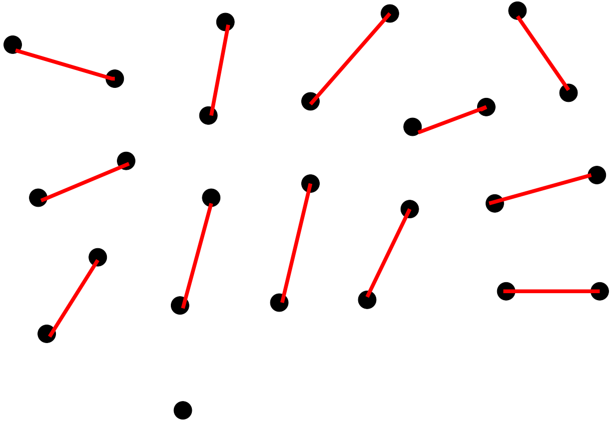
Matchings



Matchings



Matchings



Phase 1 ends and Phase 2 begins the first time that there are no vertices of degree one.

Matchings

Phase 1 ends and Phase 2 begins the first time that there are no vertices of degree one.

No mistakes are made in Phase 1 i.e. the set of edges chosen are in some largest matching.

Matchings

Phase 1 ends and Phase 2 begins the first time that there are no vertices of degree one.

No mistakes are made in Phase 1 i.e. the set of edges chosen are in some largest matching.

Karp and Sipser analysed the algorithms performance on $G_{n,p}$, where $p = c/n$.
In $G_{n,p}$ each of the $\binom{n}{2}$ edges of the complete graph K_n appear independently with probability p.
Aronson, Frieze and Pittel (1998) gave a more precise analysis.

Karp and Sipser analysed the algorithms performance on $G_{n,p}$, where $p = c/n$.
In $G_{n,p}$ each of the $\binom{n}{2}$ edges of the complete graph K_n appear independently with probability p.
Aronson, Frieze and Pittel (1998) gave a more precise analysis.

If $c < e$ then w.h.p. Phase 1 ends with a graph with $o(n)$ vertices.
If $c < e$ then w.h.p. Phase 1 ends with a graph consisting of $O(1)$ vertex disjoint cycles, in expectation.

Matchings

Karp and Sipser analysed the algorithms performance on $G_{n,p}$, where $p = c/n$.
In $G_{n,p}$ each of the $\binom{n}{2}$ edges of the complete graph K_n appear independently with probability p.
Aronson, Frieze and Pittel (1998) gave a more precise analysis.

If $c < e$ then w.h.p. Phase 1 ends with a graph with $o(n)$ vertices.
If $c < e$ then w.h.p. Phase 1 ends with a graph consisting of $O(1)$ vertex disjoint cycles, in expectation.

If $c \geq e$ then w.h.p. Phase 2 isolates $o(n)$ vertices.
If $c \geq e$ then w.h.p. Phase 2 isolates $\Theta(n^{1/5} \log^{O(1)} n)$ vertices.

Matchings

For the graph G remaining after t steps of the algorithm, let

$$\begin{aligned} v_1 &= \text{the number of vertices of degree one} \\ v &= \text{the number of vertices of degree at least two} \\ m &= \text{the number of edges} \end{aligned}$$

Matchings

For the graph G remaining after t steps of the algorithm, let

$$\begin{aligned} v_1 &= \text{the number of vertices of degree one} \\ v &= \text{the number of vertices of degree at least two} \\ m &= \text{the number of edges} \end{aligned}$$

One can show that

- The sequence v_1, v, m is a Markov chain.

Matchings

For the graph G remaining after t steps of the algorithm, let

$$v_1 = \text{the number of vertices of degree one}$$
$$v = \text{the number of vertices of degree at least two}$$
$$m = \text{the number of edges}$$

One can show that

- The sequence v_1, v, m is a Markov chain.
- At each stage G is a random graph with these parameters.

Matchings

For the graph G remaining after t steps of the algorithm, let

$$\begin{aligned} v_1 &= \text{the number of vertices of degree one} \\ v &= \text{the number of vertices of degree at least two} \\ m &= \text{the number of edges} \end{aligned}$$

One can show that

- The sequence v_1, v, m is a Markov chain.
- At each stage G is a random graph with these parameters.
- The number of vertices v_k of degree $k \geq 2$ satisfies

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

where z is the solution to

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

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

$$\mathbf{E}[v_1' - v_1] = -1 - \frac{v_1}{2m} + \frac{v^2 z^4 e^z}{(2mf)^2} - \frac{v_1 v z^2 e^z}{(2m)^2 f} + O\left(\frac{\log^2 v}{vz}\right),$$

$$\mathbf{E}[v' - v] = -1 + \frac{v_1}{2m} - \frac{v^2 z^4 e^z}{(2mf)^2} + O\left(\frac{\log^2 v}{vz}\right),$$

$$\mathbf{E}[m' - m] = -1 - \frac{vz^2 e^z}{2mf} + O\left(\frac{\log^2 v}{vz}\right).$$

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

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

$$\mathbf{E}[v_1' - v_1] = -1 - \frac{v_1}{2m} + \frac{v^2 z^4 e^z}{(2mf)^2} - \frac{v_1 v z^2 e^z}{(2m)^2 f} + O\left(\frac{\log^2 v}{vz}\right),$$

$$\mathbf{E}[v' - v] = -1 + \frac{v_1}{2m} - \frac{v^2 z^4 e^z}{(2mf)^2} + O\left(\frac{\log^2 v}{vz}\right),$$

$$\mathbf{E}[m' - m] = -1 - \frac{vz^2 e^z}{2mf} + O\left(\frac{\log^2 v}{vz}\right).$$

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

v_1, v, m closely follow the trajectory of a set of differential equations.

These equations are:

$$\frac{dv_1}{dt} = -1 - \frac{v_1}{2m} + \frac{v^2 z^4 e^z}{(2mf)^2} - \frac{v_1 v z^2 e^z}{(2m)^2 f},$$

$$\frac{dv}{dt} = -1 + \frac{v_1}{2m} - \frac{v^2 z^4 e^z}{(2mf)^2},$$

$$\frac{dm}{dt} = -1 - \frac{v z^2 e^z}{2mf}.$$

These equations are:

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

Their solution is

$$2m = \frac{n}{c} z^2,$$
$$v = n(1 - e^{-z}(1+z))\beta,$$
$$v_1 = \frac{n}{c}\left[z^2 - zc\beta(1 - e^{-z})\right],$$
$$t = \frac{n}{c}\left[c(1-\beta) - \frac{1}{2}\log^2\beta\right],$$

where $\beta e^{c\beta} = e^z$.

Super-critical case: $c > e$

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

Super-critical case: $c > e$

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

We have observed that

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

So, it is enough to show that w.h.p. $v_1 = \tilde{O}(n^{1/5})$ throughout the algorithm, for 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

Controlling v_1

We first observe that

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

Controlling v_1

We first observe that

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

Early Phase: $z \geq n^{-1/100}$.
Whp v_1 stays $\tilde{O}(z^{-2})$.

Controlling v_1

We first observe that

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

Middle Phase: $n^{-1/100} \geq z \geq n^{-1/5}$
At this point 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 $\tilde{O}(z^{-1})$.

Controlling v_1

We first observe that

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

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 square root of this.

Matchings

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

Matchings

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

Chebolu, Frieze and Melsted (2010) 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 Karp-Sipser algorithm runs in $O(n)$ time and makes $\tilde{O}(n^{1/5})$ mistakes.

Chebolu, Frieze and Melsted (2010) 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.

Mistakes are made in Phase 2 that starts with a graph distributed as $G_{\nu,\mu}^{\delta\geq 2}$ i.e. a random graph with ν vertices and μ edges and minimum degree δ at least two.

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

Chebolu, Frieze and Melsted (2010) 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.

Mistakes are made in Phase 2 that starts with a graph distributed as $G_{\nu,\mu}^{\delta\geq 2}$ i.e. a random graph with ν vertices and μ edges and minimum degree δ at least two.

The CFM algorithm pairs up the unmatched vertices and tries to find an augmenting path joining them.

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

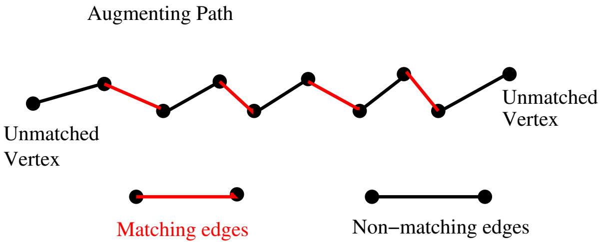
Chebolu, Frieze and Melsted (2010) show that these mistakes can be corrected i.e one can find a true maximum matching in $O(n)$ time w.h.p., for c sufficiently large.

Mistakes are made in Phase 2 that starts with a graph distributed as $G_{\nu,\mu}^{\delta\geq 2}$ i.e. a random graph with ν vertices and μ edges and minimum degree δ at least two.

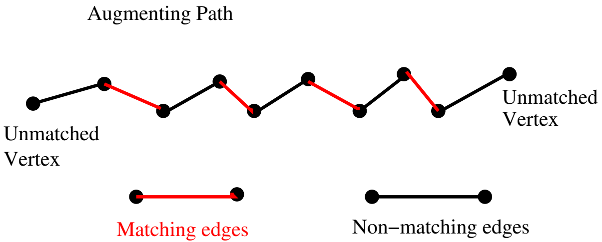
The CFM algorithm pairs up the unmatched vertices and tries to find an augmenting path joining them.

All statements from now on refer to Phase 2.

Matchings



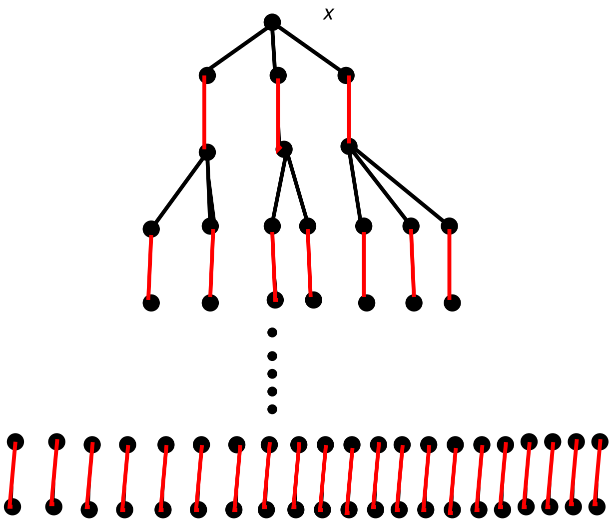
Replacing the matching edges by non-matching edges on the path, and only the path, yields a larger matching.



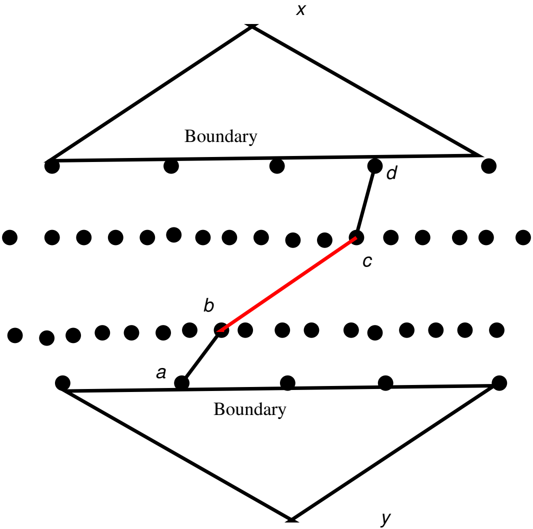
Replacing the matching edges by non-matching edges on the path, and only the path, yields a larger matching.

To find an augmenting path from unmatched vertex x to vertex unmatched vertex y, we use augmenting trees:

Matchings



Matchings



To make this work properly, we have to use all the edges of the graph at the beginning of Phase 2, even though we have "looked at" them while running KSGREEDY.

To make this work properly, we have to use all the edges of the graph at the beginning of Phase 2, even though we have "looked at" them while running KSGREEDY.

We assume that the edges of G are given to us in some fixed random order $\{e_1, e_2, \ldots, e_m\}$. When we want a random edge with a given property then we take the first edge in this order, with the required property.

Matchings

To make this work properly, we have to use all the edges of the graph at the beginning of Phase 2, even though we have "looked at" them while running KSGREEDY.

We assume that the edges of G are given to us in some fixed random order $\{e_1, e_2, \ldots, e_m\}$. When we want a random edge with a given property then we take the first edge in this order, with the required property.

If v is a vertex that is matched when there are no vertices of degree one, then we say that v is regular. The set of regular vertices is denoted by R.

To make this work properly, we have to use all the edges of the graph at the beginning of Phase 2, even though we have "looked at" them while running KSGREEDY.

We assume that the edges of G are given to us in some fixed random order $\{e_1, e_2, \ldots, e_m\}$. When we want a random edge with a given property then we take the first edge in this order, with the required property.

If v is a vertex that is matched when there are no vertices of degree one, then we say that v is regular. The set of regular vertices is denoted by R.

When a regular vertex is deleted, it will be matched to the first available edge in the order. The next edge in the order containing v is called the witness for v.

Matchings

Now fix some small $\epsilon > 0$ and another small constant α.

Matchings

Now fix some small $\epsilon > 0$ and another small constant α.

A vertex is early if it is deleted before step $n^{1-\epsilon}$ (of Phase 2) and late otherwise.

An edge e_i is punctual if $i \leq (1-\alpha)m$ and tardy otherwise.

Matchings

Now fix some small $\epsilon > 0$ and another small constant α.

A vertex is early if it is deleted before step $n^{1-\epsilon}$ (of Phase 2) and late otherwise.

An edge e_i is punctual if $i \leq (1-\alpha)m$ and tardy otherwise.

$$R_0 = \{v \in R : \ v \text{ is early and the witness of } v \text{ is punctual}\}.$$

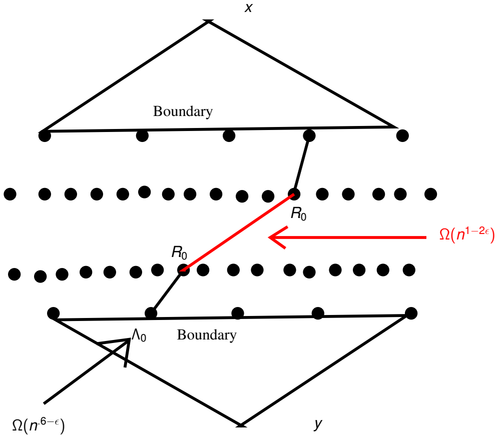
and

$$\Lambda_0 = \left\{ v : v \text{ has punctual degree at least ten in } G(n^{1-\epsilon}) \right\}$$

where $G(t)$ is the graph G after t steps of Phase 2.

The tardy $R_0 : \Lambda_0$ edges are uniformly random from $R_0 \times \Lambda_0$, conditional on all other edges.
This is because they do not affect the course of the algorithm.

These values show an expected $\Omega(n^{.2-4\epsilon})$ paths.
As such, w.h.p., we succeed in finding augmenting paths.



Contents of talk

(a) Random Discrete Structures

(b) Random Instances of the TSP in the unit square $[0, 1]^2$

(c) The Random Graphs $G_{n,m}$ and $G_{n,p}$.

  (1) Evolution
  (2) Chromatic number
  (3) Matchings
  (4) Hamilton cycles

(d) Randomly edge weighted graphs

  1. Minimum Spanning Tree
  2. Shortest Paths
  3. 3-Dimensional Assignment Problem
  4. Random Instances of the TSP with independent costs

(e) Random k-SAT

(f) Open Problems

Hamilton Cycles

Determining whether or not a graph has a Hamilton cycle is NP-hard Karp (1972).

Hamilton Cycles

Determining whether or not a graph has a Hamilton cycle is NP-hard Karp (1972).

Theorem (Komlós and Szemerédi (1983))

Suppose that $m = \frac{1}{2}n(\log n + \log\log n + c_n)$. *Then*

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

Hamilton Cycles

Determining whether or not a graph has a Hamilton cycle is NP-hard Karp (1972).

Theorem (Komlós and Szemerédi (1983))

Suppose that $m = \frac{1}{2}n(\log n + \log\log n + c_n)$. *Then*

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

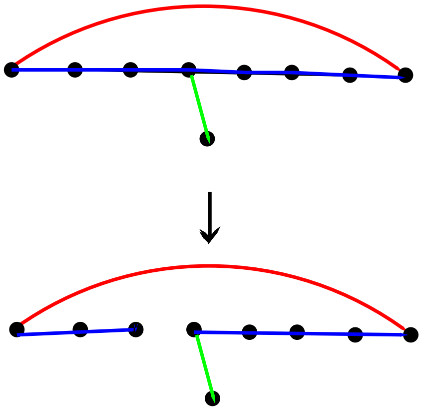
We will describe an algorithm that runs in polynomial time and finds a Hamilton cycle w.h.p. for the case $c_n = \omega \to \infty$.

Posá Rotations

We can start our algorithm with any path.

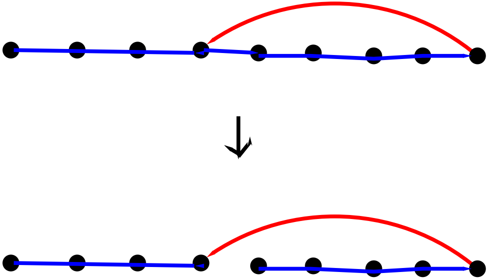
The red edge extends the blue path.

Alternative way of extending path:



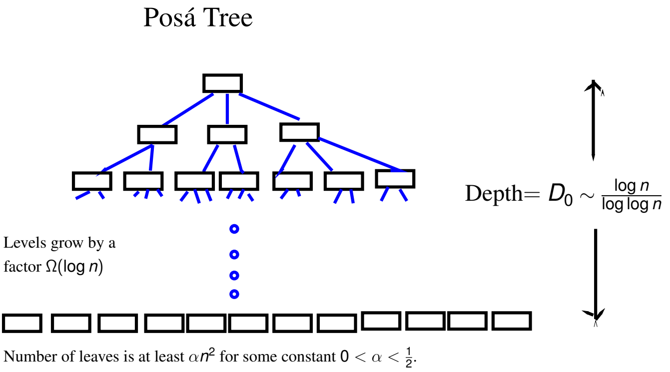
Hamilton Cycles

If there is no extension then we rotate the path:



We will in general, have several choices for the red edge here.
Each rotation gives another endpoint.

Hamilton Cycles



Each rectangle is a path that is obtained from its parent by a rotation.

BOOST is the set of pairs of endpoints in the leaves.

Hamilton Cycles

Let $m = \frac{1}{2}n(\log n + \log\log n + \omega)$ and $m_2 = \omega n/4$ and let $m_1 = m - m_2$.

Let $m = \frac{1}{2}n(\log n + \log\log n + \omega)$ and $m_2 = \omega n/4$ and let $m_1 = m - m_2$.

Choose m_2 random edges $X = \{e_1, e_2, \ldots, e_{m_2}\}$. Try to grow path using Posa trees and G_{n,m_1}.

Let $m = \frac{1}{2}n(\log n + \log\log n + \omega)$ and $m_2 = \omega n/4$ and let $m_1 = m - m_2$.

Choose m_2 random edges $X = \{e_1, e_2, \ldots, e_{m_2}\}$. Try to grow path using Posa trees and G_{n,m_1}.

If we fail to extend and grow the Posá tree to depth D_0 then we try the next edge e from X. If $e \in BOOST$ then we can extend. The probability that the next edge does this is at least A for some constant $A > 0$.

Let $m = \frac{1}{2}n(\log n + \log\log n + \omega)$ and $m_2 = \omega n/4$ and let $m_1 = m - m_2$.

Choose m_2 random edges $X = \{e_1, e_2, \ldots, e_{m_2}\}$. Try to grow path using Posa trees and G_{n,m_1}.

If we fail to extend and grow the Posá tree to depth D_0 then we try the next edge e from X. If $e \in BOOST$ then we can extend. The probability that the next edge does this is at least A for some constant $A > 0$.
So, the probability the algorithm fails can be bounded by

$$\mathbf{Pr}(Bin(m_2, A) < n) = o(1).$$

Hamilton Cycles

Let $m = \frac{1}{2}n(\log n + \log\log n + \omega)$ and $m_2 = \omega n/4$ and let $m_1 = m - m_2$.

Choose m_2 random edges $X = \{e_1, e_2, \ldots, e_{m_2}\}$. Try to grow path using Posa trees and G_{n,m_1}.

If we fail to extend and grow the Posá tree to depth D_0 then we try the next edge e from X. If $e \in BOOST$ then we can extend. The probability that the next edge does this is at least A for some constant $A > 0$.
So, the probability the algorithm fails can be bounded by

$$\mathbf{Pr}(Bin(m_2, A) < n) = o(1).$$

It is not necessary to partition the edges and the algorithm can be made deterministic, Bollobás, Fenner and Frieze (1985).

Hamilton cycles

With the threshold problem solved, existentially and constructively, we can consider other models of a random graph: We first see what happens if we condition on minimum degree at least two:

Hamilton cycles

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.

Theorem (Bollobás, Fenner and Frieze (1990))

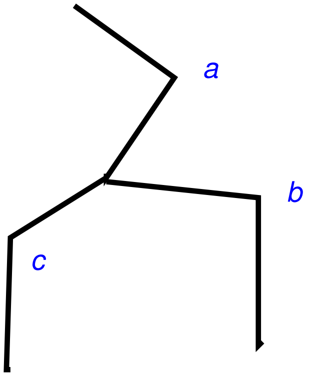
Let $m = \frac{1}{6}n(\log n + \log\log n + c_n)$ *then*

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

for some explicit function $f(c)$*.*

$e^{-f(c)}$ is the asymptotic probability that there are no spiders.

Spiders



Vertices a, b, c are of degree two

Hamilton cycles

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

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

$r = O(1)$ was proved by Robinson and Wormald (1992,1994)
$r \to \infty$ was proved by Krivelevich, Sudakov, Vu, Wormald (2001) and Cooper, Frieze, Reed (2002).

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

Theorem (Bohman and Frieze (2009))

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

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

Hamilton cycles

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

Theorem (Bohman and Frieze (2009))

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

This is not implied by the previous results on random regular graphs.
We need $k \geq 3$ to avoid spiders:

We now consider conditoning on minimum degree at least three. Let

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

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

We now consider conditoning on minimum degree at least three. Let

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

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

Conjecture true for $c = 3/2$,
Robinson and Wormald (1984)

Hamilton cycles

We now consider conditoning on minimum degree at least three. Let

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

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

Theorem (Bollobás, Cooper, Fenner, Frieze (2000))

$L_c = 1$ *for* $c \geq 128$.

We now consider conditoning on minimum degree at least three. Let

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

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

Theorem (Frieze (2012))

$L_c = 1$ *for* $c \geq 10$.

Theorem (Frieze (2012))

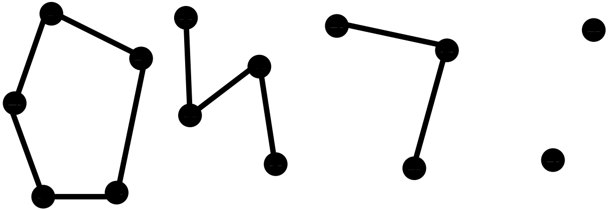
$L_c = 1$ for $c \geq 10$.

Conjecture true for $c \geq 3$. Assuming numerical solution of some differential equations.

Theorem (Frieze and Haber (2014))

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.

The improved results on Hamilton cycles in $G_{n,cn;3}$ rely on the analysis of a greedy algorithm for finding a good 2-matching M viz. a set of edges that induce a graph of maximum degree at most two.



By good, we mean that M has $O(\log n)$ components. This gives us a good basis for constructing a Hamilton cycle.
We next discuss an algorithm for finding such a 2-matching.

Algorithm 2GREEDY: The input for this algorithm is $G_{n,cn}^{\delta\geq 3}$ for c suficiently large – currently $c \geq 10$ will suffice.

Algorithm 2GREEDY: The input for this algorithm is $G_{n,cn}^{\delta\geq 3}$ for c suficiently large – currently $c \geq 10$ will suffice.

The algorithm builds a collection of paths and cycles, mainly paths. We use M to denote the set of edges chosen so far. We let B denote the set of vertices that are endpoints of paths of M.

Algorithm 2GREEDY: The input for this algorithm is $G_{n,cn}^{\delta\geq 3}$ for c suficiently large – currently $c \geq 10$ will suffice.

The algorithm builds a collection of paths and cycles, mainly paths. We use M to denote the set of edges chosen so far. We let B denote the set of vertices that are endpoints of paths of M.

In the Karp-Sipser algorithm we took care to “grab” vertices of degree one.
Here we take care to grab vertices of degree at most two if they are not incident with M and of degree one if they are.
Otherwise we choose a random edge incident to a vertex in $\bar{B}$.
We refer to these as dangerous vertices.

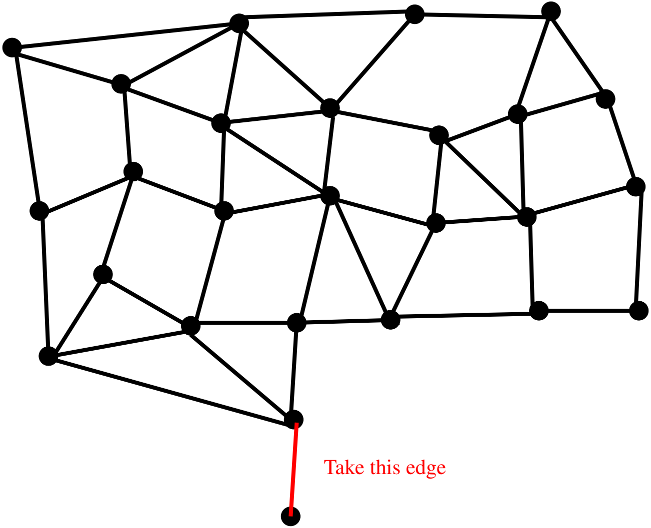
Algorithm 2GREEDY: The input for this algorithm is $G_{n,cn}^{\delta\geq 3}$ for c suficiently large – currently $c \geq 10$ will suffice.

The algorithm builds a collection of paths and cycles, mainly paths. We use M to denote the set of edges chosen so far. We let B denote the set of vertices that are endpoints of paths of M.

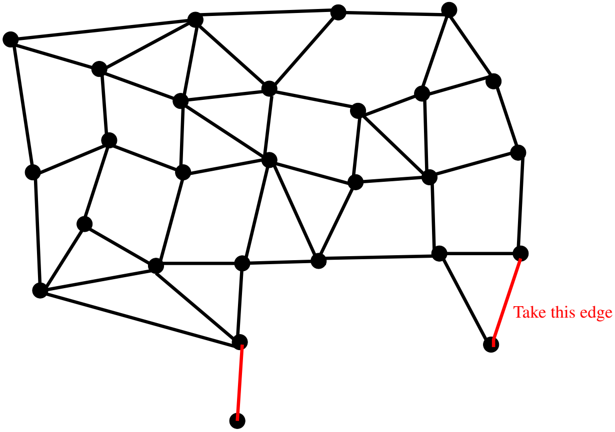
In the Karp-Sipser algorithm we took care to “grab” vertices of degree one.
Here we take care to grab vertices of degree at most two if they are not incident with M and of degree one if they are.
Otherwise we choose a random edge incident to a vertex in $\bar{B}$.
We refer to these as dangerous vertices.

Phase 1 ends when $\bar{B} = \emptyset$.

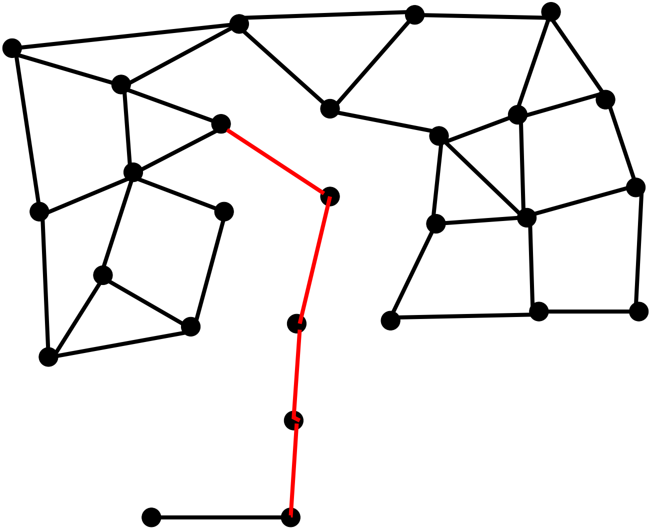
Hamilton cycles



Hamilton cycles

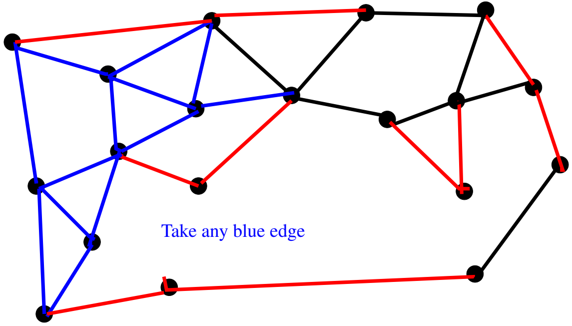


Hamilton cycles

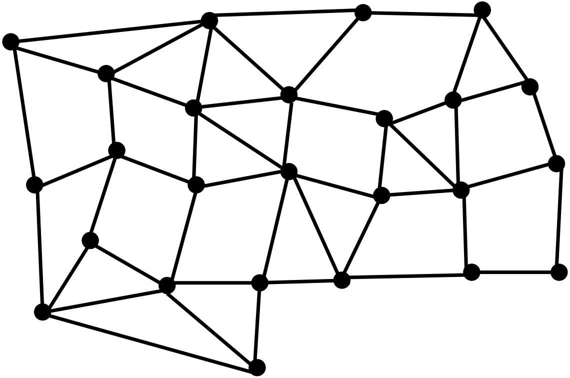


Take this edge

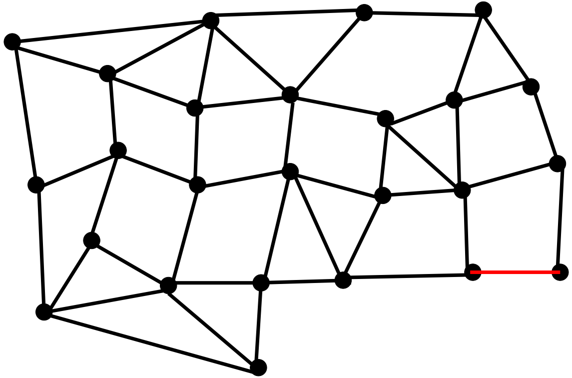
If none of these cases are applicable then we choose a random edge among those incident with a vertex not in *B* i.e. not yet covered by *M*.



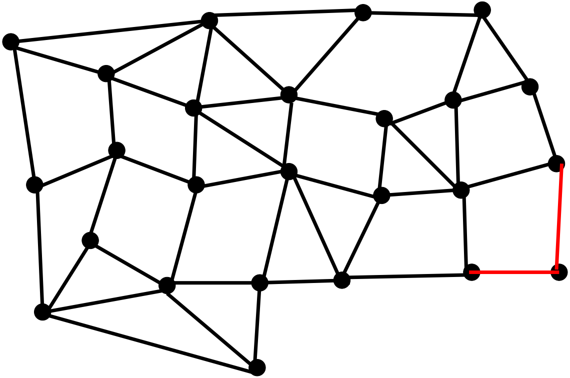
Hamilton cycles



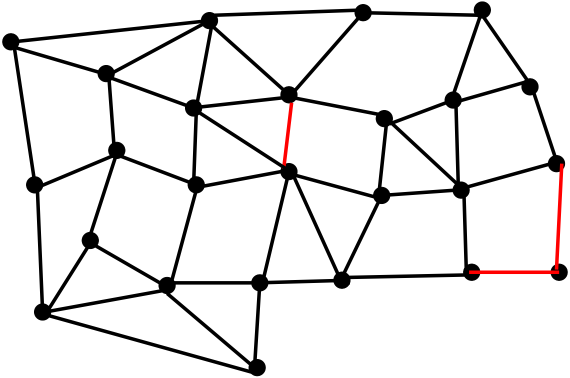
Hamilton cycles



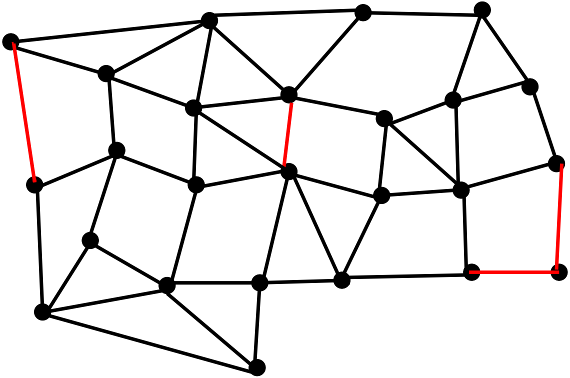
Hamilton cycles



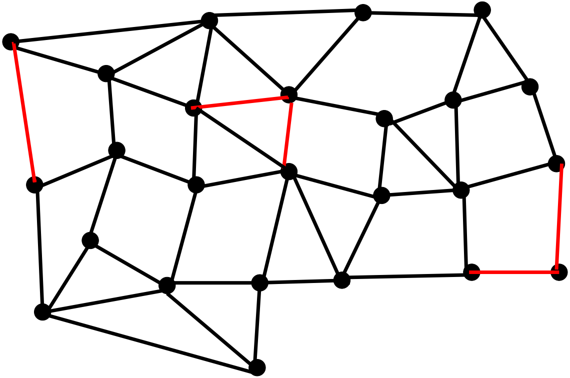
Hamilton cycles



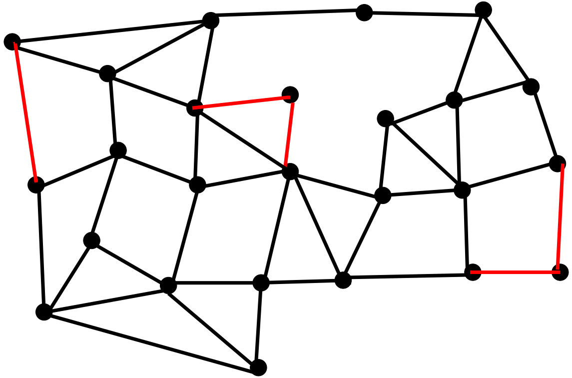
Hamilton cycles



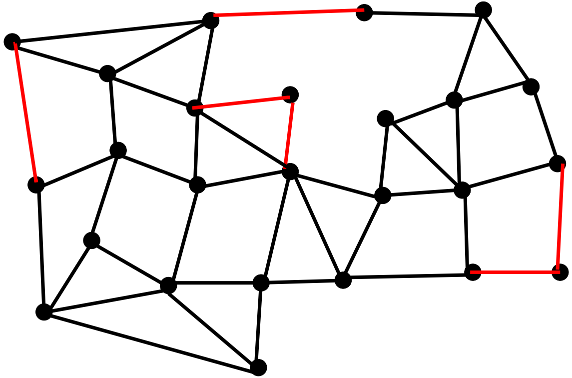
Hamilton cycles



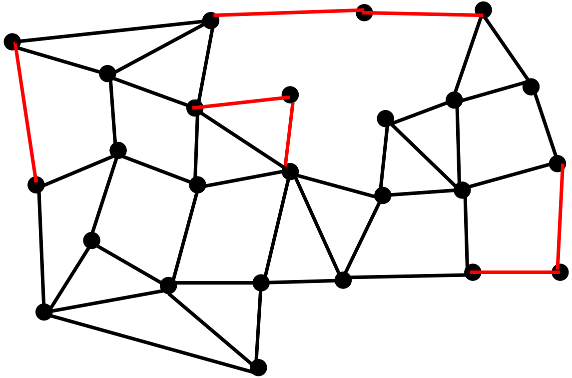
Hamilton cycles



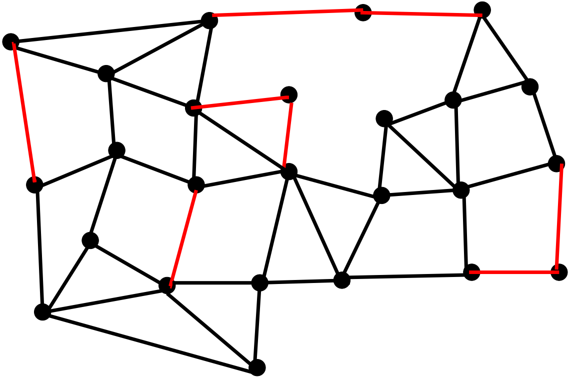
Hamilton cycles



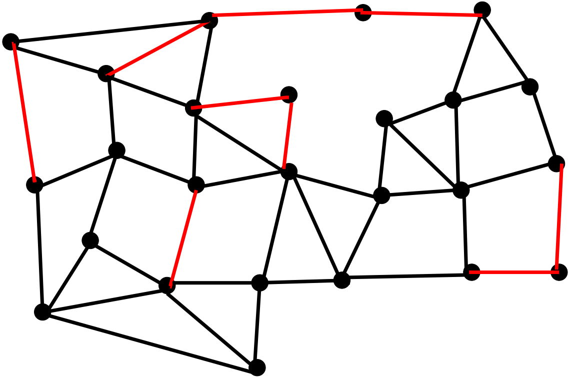
Hamilton cycles



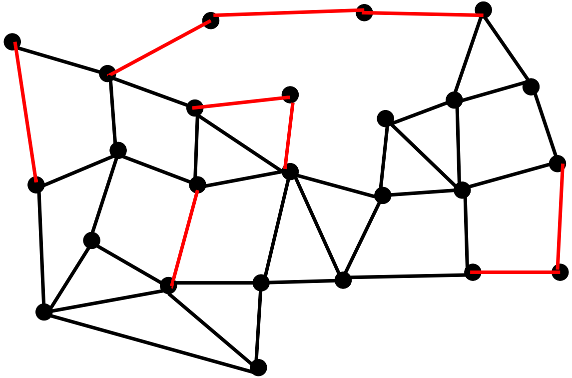
Hamilton cycles



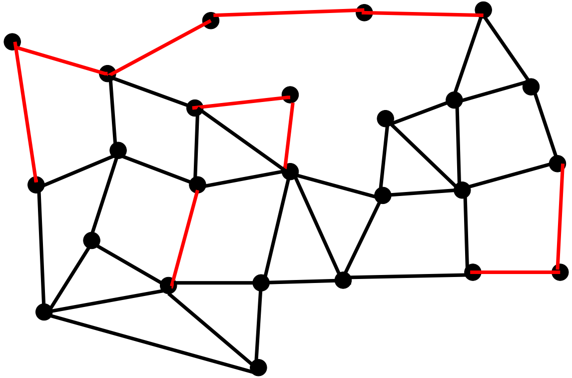
Hamilton cycles

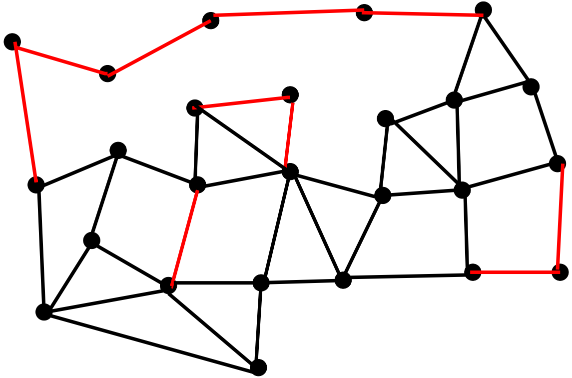


Hamilton cycles

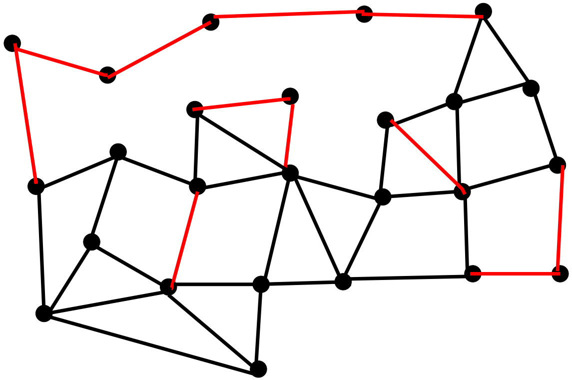


Hamilton cycles

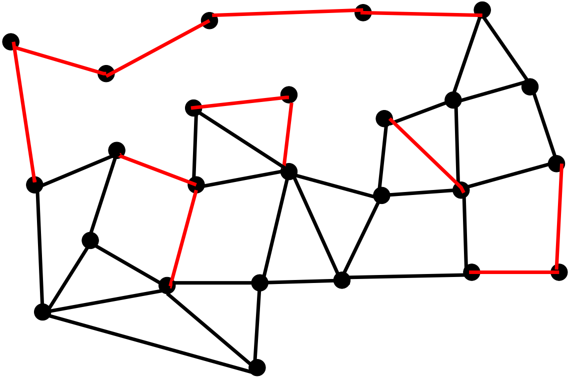




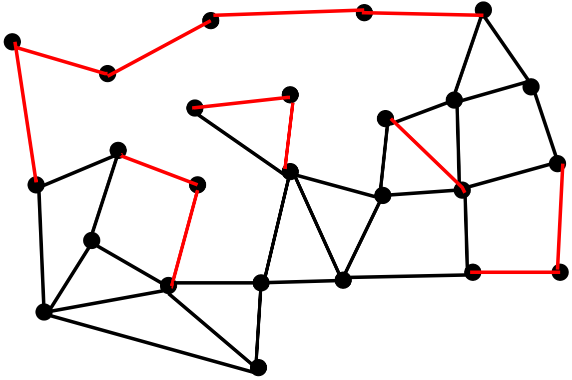
Hamilton cycles



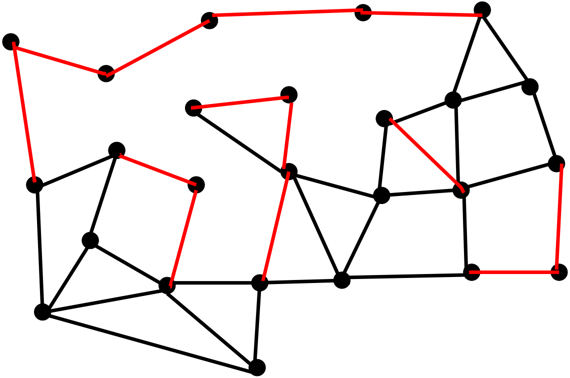
Hamilton cycles



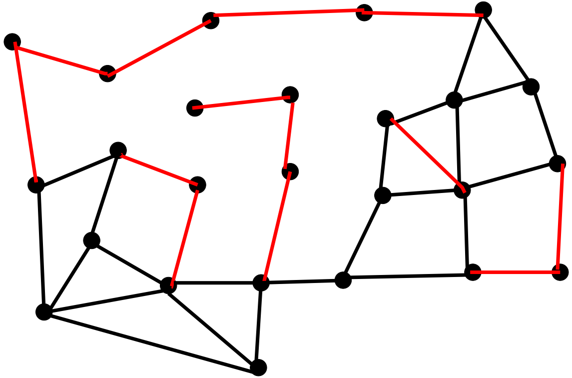
Hamilton cycles



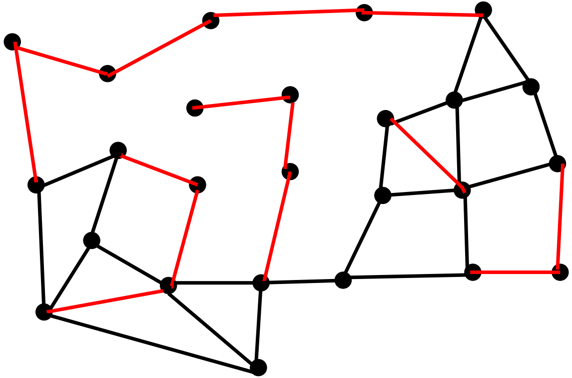
Hamilton cycles



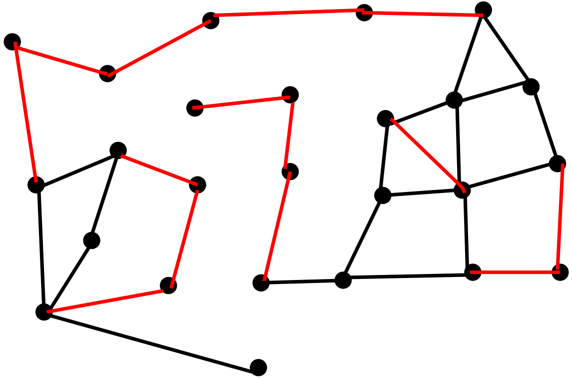
Hamilton cycles



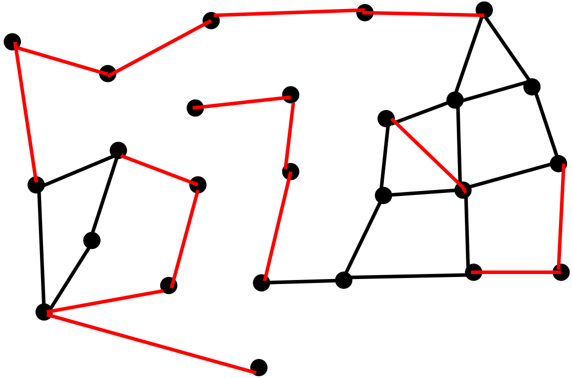
Hamilton cycles



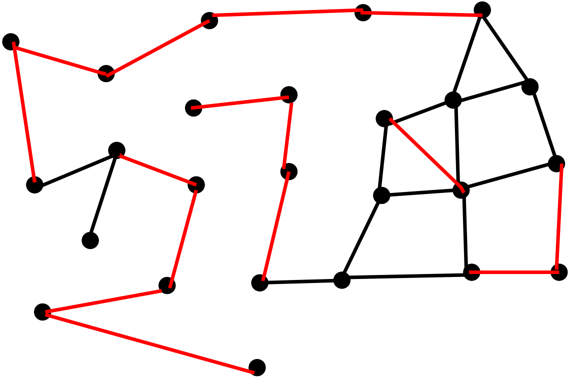
Hamilton cycles



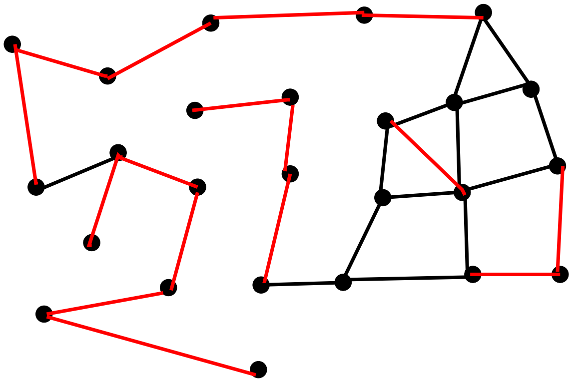
Hamilton cycles



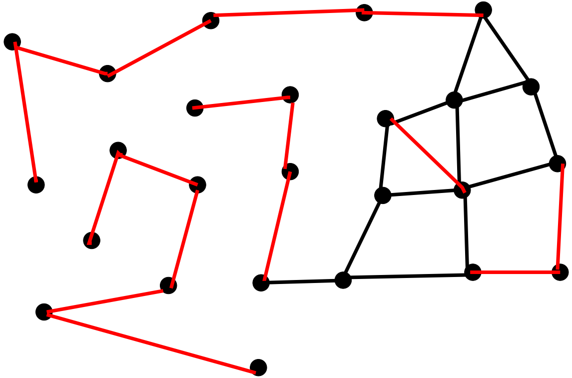
Hamilton cycles



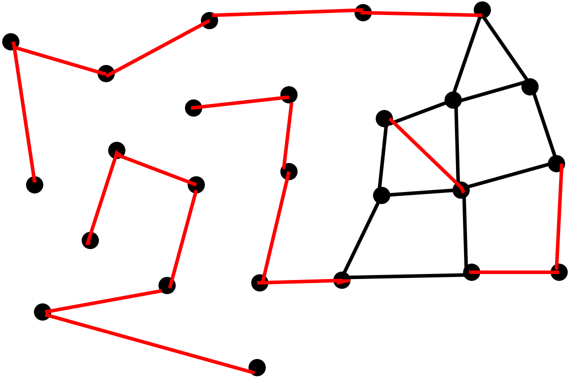
Hamilton cycles



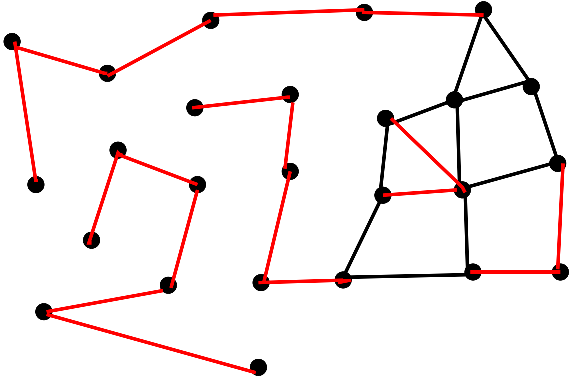
Hamilton cycles



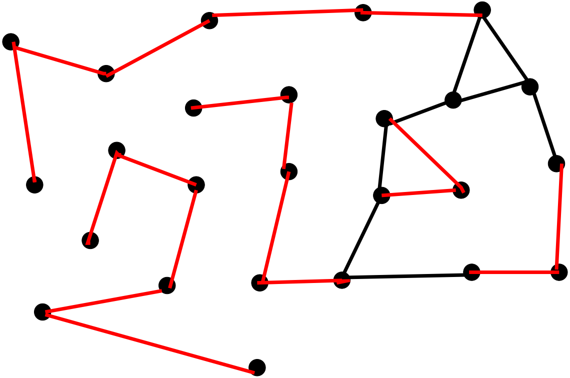
Hamilton cycles

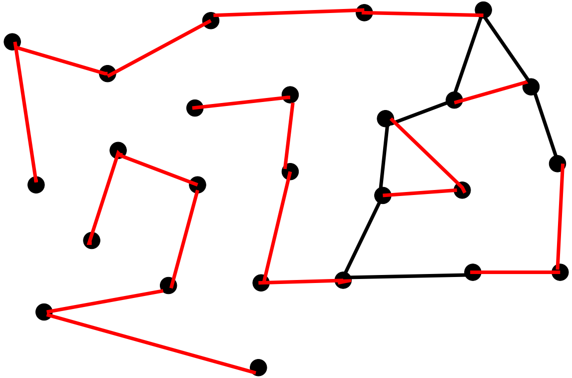


Hamilton cycles



Hamilton cycles





Phase 1 is over.

At the end of Phase 1, the 2-matching M will consist mainly of vertex disjoint paths. The isolated vertices and the cycles will play no further part in the rest of the 2GREEDY algorithm. They will be part of the output though.

We show that w.h.p. the number of paths is $\Omega(n)$ and that there are $O(\log n)$ isolated vertices.

Hamilton cycles

At the end of Phase 1, the 2-matching M will consist mainly of vertex disjoint paths. The isolated vertices and the cycles will play no further part in the rest of the 2GREEDY algorithm. They will be part of the output though.

We show that w.h.p. the number of paths is $\Omega(n)$ and that there are $O(\log n)$ isolated vertices.

The edges not in M define a graph H which is distributed as $G_{\nu,\mu}^{\delta\geq 2}$ for some $\nu, \mu = \Omega(n)$.

At the end of Phase 1, the 2-matching M will consist mainly of vertex disjoint paths. The isolated vertices and the cycles will play no further part in the rest of the 2GREEDY algorithm. They will be part of the output though.

We show that w.h.p. the number of paths is $\Omega(n)$ and that there are $O(\log n)$ isolated vertices.

The edges not in M define a graph H which is distributed as $G_{\nu,\mu}^{\delta\geq 2}$ for some $\nu, \mu = \Omega(n)$.

We find a perfect matching M' in H in $O(n)$ time. Adding M to M' produces a 2-matching in G which has $O(\log n)$ components w.h.p.

The analysis of 2-GREEDY is similar to that of the Karp-Sipser algorithm: Only, it has more parameters:

- μ is the number of edges in Γ,
- $y_k = |Y_k| = |\{v \notin B : d_\Gamma(v) = k\}|$, $k = 1, 2$,
- $z_1 = |Z_1| = |\{v \in B : d_\Gamma(v) = 1\}|$,
- $y = |Y| = |\{v \notin B : d_\Gamma(v) \geq 3\}|$.
- $z = |Z| = |\{v \in B : d_\Gamma(v) \geq 2\}|$.

The analysis of 2-GREEDY is similar to that of the Karp-Sipser algorithm: Only, it has more parameters:

- μ is the number of edges in Γ,
- $y_k = |Y_k| = |\{v \notin B : d_\Gamma(v) = k\}|, k = 1, 2,$
- $z_1 = |Z_1| = |\{v \in B : d_\Gamma(v) = 1\}|,$
- $y = |Y| = |\{v \notin B : d_\Gamma(v) \geq 3\}|.$
- $z = |Z| = |\{v \in B : d_\Gamma(v) \geq 2\}|.$

We can show that y_1, y_2, z_1 remain $O(\log n)$ throughout, w.h.p. And that Phase 1 ends with $y = 0$ and $z_1 = \Omega(n)$.

As 2-GREEDY progresses the random sequence $(y_0, y_1, y_2, z_1, y, z, \mu)$ is a Markov Chain.

The graph defined by the remaining vertices and edges is chosen uniformly from the set of graphs with these parameters.

The degrees of vertices in Y, Z are close to truncated Poisson:

Let $f_i(x) = e^x - \sum_{t=0}^{i-1} \frac{x^t}{t!}$ and let λ be the solution to

$$\frac{y\lambda f_2(\lambda)}{f_3(\lambda)} + \frac{z\lambda f_1(\lambda)}{f_2(\lambda)} = 2\mu - y_1 - 2y_2 - z_1.$$

Then w.h.p.

$$y_k \sim \frac{y\lambda^k}{k! f_3(\lambda)}, k \geq 3 \text{ and } z_k \sim \frac{z\lambda^k}{k! f_2(\lambda)}, k \geq 2.$$

There are differential equations that closely model the process: They only involve variables $\hat{y}, \hat{z}, \hat{\mu}$ that represent y, z, μ, other variables stay small.
It takes some effort, but we reduce them to

There are differential equations that closely model the process: They only involve variables $\hat{y}, \hat{z}, \hat{\mu}$ that represent y, z, μ, other variables stay small.
It takes some effort, but we reduce them to

$$\frac{d\hat{y}}{dt} = \hat{A} + \hat{B} - \hat{C} - 1; \quad \frac{d\hat{z}}{dt} = 2\hat{C} - 2\hat{A} - 2\hat{B}; \quad \frac{d\hat{\mu}}{dt} = -1 - \hat{D}$$

where

$$\hat{A} = \frac{\hat{y}\hat{z}\hat{\lambda}^5 f_0(\hat{\lambda})}{8\hat{\mu}^2 f_2(\hat{\lambda}) f_3(\hat{\lambda})}, \ \hat{B} = \frac{\hat{z}^2\hat{\lambda}^4 f_0(\hat{\lambda})}{4\hat{\mu}^2 f_2(\hat{\lambda})^2}, \ \hat{C} = \frac{\hat{y}\hat{\lambda} f_2(\hat{\lambda})}{2\hat{\mu} f_3(\hat{\lambda})}, \ \hat{D} = \frac{\hat{z}\hat{\lambda}^2 f_0(\hat{\lambda})}{2\hat{\mu} f_2(\hat{\lambda})}$$

and

$$\frac{\hat{y}\hat{\lambda} f_2(\hat{\lambda})}{f_3(\hat{\lambda})} + \frac{\hat{z}\hat{\lambda} f_1(\hat{\lambda})}{f_2(\hat{\lambda})} = 2\hat{\mu}.$$

There are differential equations that closely model the process: They only involve variables $\hat{y}, \hat{z}, \hat{\mu}$ that represent y, z, μ, other variables stay small.
It takes some effort, but we reduce them to

$$\frac{d\hat{y}}{dt} = \hat{A} + \hat{B} - \hat{C} - 1; \quad \frac{d\hat{z}}{dt} = 2\hat{C} - 2\hat{A} - 2\hat{B}; \quad \frac{d\hat{\mu}}{dt} = -1 - \hat{D}$$

where

$$\hat{A} = \frac{\hat{y}\hat{z}\hat{\lambda}^5 f_0(\hat{\lambda})}{8\hat{\mu}^2 f_2(\hat{\lambda}) f_3(\hat{\lambda})}, \quad \hat{B} = \frac{\hat{z}^2\hat{\lambda}^4 f_0(\hat{\lambda})}{4\hat{\mu}^2 f_2(\hat{\lambda})^2}, \quad \hat{C} = \frac{\hat{y}\hat{\lambda} f_2(\hat{\lambda})}{2\hat{\mu} f_3(\hat{\lambda})}, \quad \hat{D} = \frac{\hat{z}\hat{\lambda}^2 f_0(\hat{\lambda})}{2\hat{\mu} f_2(\hat{\lambda})}$$

and

$$\frac{\hat{y}\hat{\lambda} f_2(\hat{\lambda})}{f_3(\hat{\lambda})} + \frac{\hat{z}\hat{\lambda} f_1(\hat{\lambda})}{f_2(\hat{\lambda})} = 2\hat{\mu}.$$

Unfortunately, we have not been able to solve these equations.

We observe however, that if $\hat{\lambda}$ is large then

$$\hat{A} \ll 1; \quad \hat{B} \ll 1; \quad \hat{C} \approx \frac{\hat{y}\hat{\lambda}}{2\hat{\mu}}; \quad \hat{D} \approx \frac{\hat{z}\hat{\lambda}^2}{2\hat{\mu}}; \quad \hat{\lambda} \approx \frac{2\hat{\mu}}{\hat{y}+\hat{z}}.$$

They can then be approximated by the following equations:

$$\begin{aligned}
\tilde{y}' &= -\frac{\tilde{y}}{\tilde{y}+\tilde{z}} - 1\\
\tilde{z}' &= \frac{2\tilde{y}}{\tilde{y}+\tilde{z}}\\
\tilde{\mu}' &= -1 - \frac{2\tilde{z}\tilde{\mu}}{(\tilde{y}+\tilde{z})^2}\\
\tilde{\lambda} &= \frac{2\tilde{\mu}}{\tilde{y}+\tilde{z}}.
\end{aligned}$$

They can then be approximated by the following equations:

$$\tilde{y}' = -\frac{\tilde{y}}{\tilde{y}+\tilde{z}} - 1$$
$$\tilde{z}' = \frac{2\tilde{y}}{\tilde{y}+\tilde{z}}$$
$$\tilde{\mu}' = -1 - \frac{2\tilde{z}\tilde{\mu}}{(\tilde{y}+\tilde{z})^2}$$
$$\tilde{\lambda} = \frac{2\tilde{\mu}}{\tilde{y}+\tilde{z}}.$$

These are solvable and they have the property that there is a time $\tilde{T}$ such that $\tilde{y}(\tilde{T}) = 0$ and $\tilde{z}(\tilde{T}) = \Omega(n)$.

These are solvable and they have the property that there is a time $\tilde{T}$ such that $\tilde{y}(\tilde{T}) = 0$ and $\tilde{z}(\tilde{T}) = \Omega(n)$.

When $c \geq 10$ we can use this to show that there is a time $\hat{T}$ such that $\hat{y}(\hat{T}) = 0$ and $\hat{z}(\hat{T}) = \Omega(n)$.

These are solvable and they have the property that there is a time $\tilde{T}$ such that $\tilde{y}(\tilde{T}) = 0$ and $\tilde{z}(\tilde{T}) = \Omega(n)$.

When $c \geq 10$ we can use this to show that there is a time $\hat{T}$ such that $\hat{y}(\hat{T}) = 0$ and $\hat{z}(\hat{T}) = \Omega(n)$.

And then because the diferential equations describe the process very closely, we can deduce that w.h.p. there is a time T such that $y_1(T) = y_2(T) = z_1(T) = \zeta(T) = y(T) = 0$ and $z(T) = \Omega(n)$.

These are solvable and they have the property that there is a time $\tilde{T}$ such that $\tilde{y}(\tilde{T}) = 0$ and $\tilde{z}(\tilde{T}) = \Omega(n)$.

When $c \geq 10$ we can use this to show that there is a time $\hat{T}$ such that $\hat{y}(\hat{T}) = 0$ and $\hat{z}(\hat{T}) = \Omega(n)$.

And then because the diferential equations describe the process very closely, we can deduce that w.h.p. there is a time T such that $y_1(T) = y_2(T) = z_1(T) = \zeta(T) = y(T) = 0$ and $z(T) = \Omega(n)$.

In which case, as already noted, we will end Phase 1 having only isolated $O(\log n)$ vertices.

Hamilton cycles

Numerical experiments suggest that such a T exists for $c \geq 2.5$, maybe even for smaller c.

c	y_{final}	z_{final}	μ_{final}	λ_{final}
3.0	0.000008	0.283721	0.398527	1.822428
2.9	0.000009	0.242563	0.326139	1.602749
2.8	0.000010	0.197461	0.253645	1.370798
2.7	0.000010	0.148901	0.182327	1.123928
2.6	0.000010	0.098344	0.114494	0.858355
2.5	0.000010	0.048976	0.054010	0.565840

These are the results of running Euler's method with step length 10^{-5} on the differential equations.

Converting the 2-matching to a hamilton cycle.
A regular vertex v is one that is deleted when there are no dangerous vertices to grab. The set of regular vertices is denoted by R.

Hamilton cycles

Converting the 2-matching to a hamilton cycle.
A regular vertex v is one that is deleted when there are no dangerous vertices to grab. The set of regular vertices is denoted by R.

When a regular vertex is deleted, it will be matched to the first available edge in the order. The next edge in the order containing v is called the witness for v.

Converting the 2-matching to a hamilton cycle.
A regular vertex v is one that is deleted when there are no dangerous vertices to grab. The set of regular vertices is denoted by R.

When a regular vertex is deleted, it will be matched to the first available edge in the order. The next edge in the order containing v is called the witness for v.

We define R_0, Λ_0 more or less as before

$$R_0 = \{v \in R : \; v \text{ is early and the witness of } v \text{ is punctual}\}.$$
$$\Lambda_0 = \left\{v : v \text{ has punctual degree at least ten in } G(n^{1-o(1)})\right\}$$

Once again, the tardy $R_0 : \Lambda_0$ edges are uniformly random from $R_0 \times \Lambda_0$, conditional on all other edges.

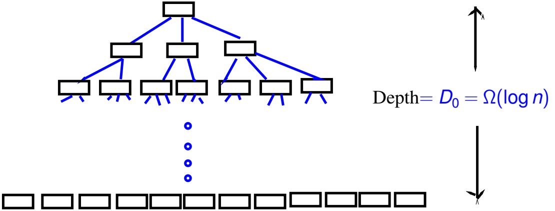
Hamilton cycles

We start our search for a Hamilton cycle by choosing a longest path in the 2-matching M.

We start our search for a Hamilton cycle by choosing a longest path in the 2-matching M.

We try to grow our path using extensions and rotations. With a given path P with endpoints v, w we grow a Posa tree with v as one endpint of all the paths produced.

Posá Tree: rotations with one endpoint fixed.



Number of leaves in Λ_0 is at least $n^{1/2+o(1)}$.

In the above diagram, each rectangle is a path that is obtained

We let *END* denote the set of endpoints of the paths produced, other than v.

We let *END* denote the set of endpoints of the paths produced, other than v.

We either manage an extension of the current path or we grow *END* to size $n^{1/2+o(1)}$.

We let *END* denote the set of endpoints of the paths produced, other than v.

We either manage an extension of the current path or we grow *END* to size $n^{1/2+o(1)}$.

For each $x \in END$ we carry out the same, creating a set of paths with $n^{1/2+o(1)}$ endpoints $END(x)$.

We let *END* denote the set of endpoints of the paths produced, other than v.

We either manage an extension of the current path or we grow *END* to size $n^{1/2+o(1)}$.

For each $x \in END$ we carry out the same, creating a set of paths with $n^{1/2+o(1)}$ endpoints $END(x)$.

We argue that w.h.p. all paths produced contain $n^{1-o(1)}$ members of R_0.

We let END denote the set of endpoints of the paths produced, other than v.

We either manage an extension of the current path or we grow END to size $n^{1/2+o(1)}$.

For each $x \in END$ we carry out the same, creating a set of paths with $n^{1/2+o(1)}$ endpoints $END(x)$.

We argue that w.h.p. all paths produced contain $n^{1-o(1)}$ members of R_0.

if we fail to extend, then the probability we fail to find a tardy $R_0 : \Lambda_0$ edge joining $x \in END$ to $y \in END(x)$ is $n^{-o(1)}$.

We only need to extend/close a cycle $O(\log^2 n)$ times and so the probability we fail is $O(n^{-o(1)} \log^2 n) = o(1)$, if we are careful with our $o(1)$'s.

So, for c sufficiently large we can find a Hamilton cycle in $G_{n,cn}^{\delta \geq 3}$ in $O(n^{1+o(1)})$ time.

Contents of talk

(a) Random Discrete Structures

(b) Random Instances of the TSP in the unit square $[0,1]^2$

(c) The Random Graphs $G_{n,m}$ and $G_{n,p}$.

  (1) Evolution
  (2) Chromatic number
  (3) Matchings
  (4) Hamilton cycles

(d) Randomly edge weighted graphs

  1. Minimum Spanning Tree
  2. Shortest Paths
  3. 3-Dimensional Assignment Problem
  4. Random Instances of the TSP with independent costs

(e) Random k-SAT

(f) Open Problems

Contents of talk

(a) Random Discrete Structures

(b) Random Instances of the TSP in the unit square $[0,1]^2$

(c) The Random Graphs $G_{n,m}$ and $G_{n,p}$.

  (1) Evolution
  (2) Chromatic number
  (3) Matchings
  (4) Hamilton cycles

(d) Randomly edge weighted graphs

  1. Minimum Spanning Tree
  2. Shortest Paths
  3. 3-Dimensional Assignment Problem
  4. Random Instances of the TSP with independent costs

(e) Random k-SAT

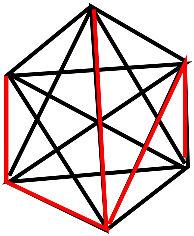
(f) Open Problems

Minimum Length Spanning Tree

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.

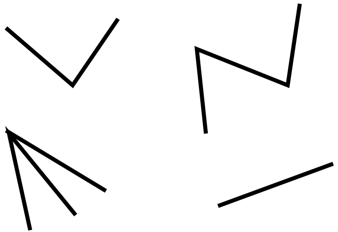
Z_n is the minimum total length of a spanning tree i.e. a connected subgraph that contains $n - 1$ edges and no cycles.



Spanning Tree

Length=sum of lengths of edges.

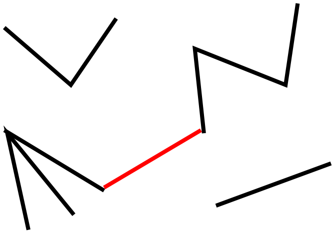
Greedy Algorithm



F is the forest induced by the edges chosen so far.

F has 4 components.

Greedy Algorithm



F is the forest induced
by the edges chosen so far.

Edges between components are longer
than edges inside components

The algorithm adds the shortest edge joining components of F.

The algorithm adds longer and longer edges as it progresses.

Minimum Length Spanning Tree

If ℓ_F is the length of the longest edge in F, then edges of length at most ℓ are contained in the components of F.

Therefore the algorithm adds $\kappa - 1$ more edges, where κ is the number of components in the graph spanned by edges of length at most ℓ_F.

Minimum Length Spanning Tree

Let T be the minimum spanning tree and let ℓ denote length.

$$\begin{aligned} Z_n = \ell(T) &= \sum_{e \in T} X_e \\ &= \sum_{e \in T} \int_{p=0}^{1} 1_{(p \le X_e)} dp \\ &= \int_{p=0}^{1} \sum_{e \in T} 1_{(p \le X_e)} dp \\ &= \int_{p=0}^{1} |\{e \in T : p \le X_e\}| dp \end{aligned}$$

$$\ell(T) = \int_{p=0}^{1} |\{e \in T : X_e \geq p\}| dp$$
$$= \int_{p=0}^{1} (\kappa(G_p) - 1) dp,$$

where $\kappa(G_p)$ is the number of components in the graph induced by edges of length at most p.

$$\ell(T) = \int_{p=0}^{1} |\{e \in T : X_e \geq p\}| dp$$
$$= \int_{p=0}^{1} (\kappa(G_p) - 1) dp,$$

where $\kappa(G_p)$ is the number of components in the graph induced by edges of length at most p.

So

$$\mathbf{E}(Z_n) = \int_{p=0}^{1} (\mathbf{E}(\# \text{ components in } G_{n,p}) - 1) dp.$$

Minimum Length Spanning Tree

$$\mathbf{E}(Z_n) = \int_{p=0}^{1} (\mathbf{E}(\#\text{ components in } G_{n,p}) - 1)dp.$$

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,

$$\mathbf{E}(Z_n) \sim \int_{p=0}^{6 \log n/n} \mathbf{E}(\#\text{ small isolated trees in } G_{n,p})dp.$$

Minimum Length Spanning Tree

$$\mathbf{E}(Z_n) \sim \int_{p=0}^{6\log n/n} \mathbf{E}(\#\text{ small isolated trees in } G_{n,p})dp$$

$$\sim \int_{p=0}^{6\log n/n} \left(\sum_{k=1}^{\log^2 n} \binom{n}{k} k^{k-2} p^{k-1} (1-p)^{k(n-k)+\binom{k}{2}-k+1} \right) dp$$

$$\sim \sum_{k=1}^{\log^2 n} \frac{n^k}{k!} k^{k-2} \frac{k!(k(n-k))!}{(k(n-k+1))!}$$

Minimum Length Spanning Tree

$$\mathbf{E}(Z_n) \sim \int_{p=0}^{6\log n/n} \mathbf{E}(\#\text{ small isolated trees in } G_{n,p})dp$$
$$\sim \int_{p=0}^{6\log n/n} \left(\sum_{k=1}^{\log^2 n} \binom{n}{k} k^{k-2} p^{k-1} (1-p)^{k(n-k)+\binom{k}{2}-k+1} \right) dp$$
$$\sim \sum_{k=1}^{\log^2 n} \frac{n^k}{k!} k^{k-2} \frac{k!(k(n-k)!}{(k(n-k+1)!}$$

So,

$$\mathbf{E}(Z_n) \sim \sum_{k=1}^{\log^2 n} \frac{1}{k^3} \sim \zeta(3).$$

Minimum Length Spanning Tree

This is most of the proof of the following:

Theorem (Frieze (1985))

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

Original proof not so "clean":
Remarkable integral formula is due to Janson (1995).

Minimum Length Spanning Tree

This is most of the proof of the following:

Theorem (Frieze (1985))

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

Original proof not so "clean":
Remarkable integral formula is due to Janson (1995).

With more work we have

Theorem (Cooper, Frieze, Ince, Janson, Spencer (2014))

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

Theorem (Cooper, Frieze, Ince, Janson, Spencer (2014))

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

$$c_1 = -1 - \zeta(3) - \frac{1}{2}\int_{x=0}^{\infty} \log\left(1-(1+x)e^{-x}\right)\,dx$$

and

$$c_2 = \int_{x=0}^{\infty}\left(x^{-3}\psi(x^{3/2})e^{-x^3/24} - x^{-3} - \sqrt{\frac{\pi}{8}}x^{-3/2} - \frac{1}{2}\right)\,dx$$

where if $\mathcal{B}_{\mathrm{ex}} = \int_{s=0}^{1} B_{\mathrm{ex}}(s)\,ds$ is the area under a normalized Brownian excursion,

$$\psi(t) = \mathbf{E}e^{t\mathcal{B}_{\mathrm{ex}}},$$

the moment generating function ψ of $\mathcal{B}_{\mathrm{ex}}$.

Minimum Length Spanning Tree

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

Theorem (Beveridge, Frieze, McDiarmid (1998))

$$\mathbf{E}(Z_n) = \frac{n}{r}(\zeta(3) + \epsilon_r)$$

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

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

Contents of talk

(a) Random Discrete Structures

(b) Random Instances of the TSP in the unit square $[0,1]^2$

(c) The Random Graphs $G_{n,m}$ and $G_{n,p}$.

  (1) Evolution
  (2) Chromatic number
  (3) Matchings
  (4) Hamilton cycles

(d) Randomly edge weighted graphs

  1. Minimum Spanning Tree
  2. Shortest Paths
  3. 3-Dimensional Assignment Problem
  4. Random Instances of the TSP with independent costs

(e) Random k-SAT

(f) Open Problems

Shortest Paths

Every edge e of the complete graph K_n is given a random length X_e.

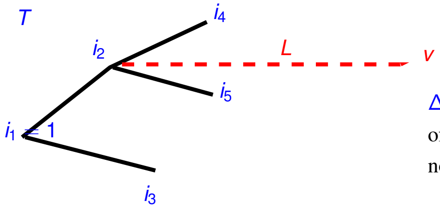
The edge lengths are independently exponentially distributed with mean 1 viz. $E(1)$ i.e. $\mathbf{Pr}(X_e \geq \lambda) = e^{-\lambda}$.

The length of a path is the sum of the lengths of its edges.

The question to be discussed is what is the length of a shortest path between two given vertices.

Let D_i be the length of a shortest path from vertex 1 to vertex i.

We can build a tree of shortest paths, adding the next closest vertex to 1 in each step.

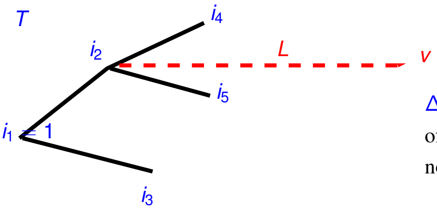


Paths in tree T are shortest paths.

Δ_v is the minimum length of a path to $v \notin T$ using one edge not in T.

If $\Delta_w = \min \Delta_v,\ v \notin T$ then $D_w = \Delta_w$.

In the above diagram we have added the 4 closest vertices to create a tree T. To find the 5th closest we compute Δ_v for each $v \notin T$ and then add the vertex that minimises Δ.



Δ_v is the minimum length of a path to $v \notin T$ using one edge not in T.

If L is the length of (i_2, v) then L is exponential conditioned on $D_{i_2} + L \geq D_{i_5}$.

So $D_{i_2} + L = D_{i_5} + E(1)$.
(Memoryless property of exponential).

So, if we add vertices to T in the order $i_1 = 1, i_2, \ldots, i_n$ then $D_{i_{k+1}} - D_{i_k}$ is the minimum of $k(n-k)$ independent $E(1)$'s.

So, if Z_i is the distance from 1 to the ith closest vertex,

$$Z_1 = 0 \text{ and } \mathbf{E}(Z_{k+1}) = \mathbf{E}(Z_k) + \frac{1}{k(n-k)}.$$

It follows that

$$\mathbf{E}(Z_n) = \frac{2}{n}\sum_{i=1}^{n-1}\frac{1}{i}.$$

Furthermore, 2 is equally likely to be the ith closest, for $i = 2, 3, \ldots, n$ and we have

$$\mathbf{E}(D_2) = \frac{1}{n-1}\sum_{i=1}^{n-1}\frac{1}{i}.$$

Theorem (Janson (1999))

Let $D_{i,j}$ be the shortest distance between i, j in the above model. Then

$$D_{1,2} \sim \frac{\log n}{n}.$$

$$\max_j D_{1,j} \sim \frac{2\log n}{n}.$$

$$\max_{i,j} D_{i,j} \sim \frac{3\log n}{n}.$$

Contents of talk

(a) Random Discrete Structures
(b) Random Instances of the TSP in the unit square $[0,1]^2$
(c) The Random Graphs $G_{n,m}$ and $G_{n,p}$.
  (1) Evolution
  (2) Chromatic number
  (3) Matchings
  (4) Hamilton cycles
(d) Randomly edge weighted graphs
  1. Minimum Spanning Tree
  2. Shortest Paths
  3. 3-Dimensional Assignment Problem
  4. Random Instances of the TSP with independent costs
(e) Random k-SAT
(f) Open Problems

Background: Two-dimensional Assignment problem

Let M be a real $n \times n$ matrix of costs.

Problem: Minimise

$$\sum_{i=1}^{n} \sum_{j=1}^{n} M_{i,j} x_{i,j} \qquad \text{subject to}$$

$$\sum_{i=1}^{n} x_{i,j} = 1, \qquad j \in [n]$$

$$\sum_{j=1}^{n} x_{i,j} = 1, \qquad i \in [n]$$

$$x_{i,j} \in \{0, 1\}$$

Solvable in polynomial ($O(n^3)$) time.

Suppose now that M is a matrix of i.i.d. random variables: Let

$Z_A = Z_A(n)$ denote the random minimum value.

(a) $\mathbf{E}(Z_A) \leq 3$ – Walkup (1979) ($M_{i,j}$ is uniform $[0, 1]$).

Suppose now that M is a matrix of i.i.d. random variables: Let

$Z_A = Z_A(n)$ denote the random minimum value.

(a) $\mathbf{E}(Z_A) \leq 3$ – Walkup (1979) ($M_{i,j}$ is uniform $[0,1]$).

(b) $\mathbf{E}(Z_A) \leq 2$ – Karp (1983) ($M_{i,j}$ is uniform $[0,1]$)

Suppose now that M is a matrix of i.i.d. random variables: Let

$Z_A = Z_A(n)$ denote the random minimum value.

(a) $\mathbf{E}(Z_A) \leq 3$ – Walkup (1979) ($M_{i,j}$ is uniform $[0, 1]$).

(b) $\mathbf{E}(Z_A) \leq 2$ – Karp (1983) ($M_{i,j}$ is uniform $[0, 1]$)

(c) $\lim_{n\to\infty} \mathbf{E}(Z_A)$ exists – Aldous (1992) ($M_{i,j}$ is $Exp(1)$)

Suppose now that M is a matrix of i.i.d. random variables: Let

$Z_A = Z_A(n)$ denote the random minimum value.

(a) $\mathbf{E}(Z_A) \leq 3$ – Walkup (1979) ($M_{i,j}$ is uniform $[0,1]$).

(b) $\mathbf{E}(Z_A) \leq 2$ – Karp (1983) ($M_{i,j}$ is uniform $[0,1]$)

(c) $\lim_{n\to\infty} \mathbf{E}(Z_A)$ exists – Aldous (1992) ($M_{i,j}$ is $Exp(1)$)

(d) $\lim_{n\to\infty} \mathbf{E}(Z_A) = \zeta(2) = \frac{\pi^2}{6}$ – Aldous (2001) ($M_{i,j}$ is $Exp(1)$)

Suppose now that M is a matrix of i.i.d. random variables: Let

$Z_A = Z_A(n)$ denote the random minimum value.

(a) $\mathbf{E}(Z_A) \leq 3$ – Walkup (1979) ($M_{i,j}$ is uniform $[0,1]$).

(b) $\mathbf{E}(Z_A) \leq 2$ – Karp (1983) ($M_{i,j}$ is uniform $[0,1]$)

(c) $\lim_{n\to\infty} \mathbf{E}(Z_A)$ exists – Aldous (1992) ($M_{i,j}$ is $Exp(1)$)

(d) $\lim_{n\to\infty} \mathbf{E}(Z_A) = \zeta(2) = \frac{\pi^2}{6}$ – Aldous (2001) ($M_{i,j}$ is $Exp(1)$)

(e) $\mathbf{E}(Z_A(n)) = \sum_{k=1}^{n} \frac{1}{k^2}$ – Linusson and Wästlund (2004) and Nair, Prabhakar and Sharmar (2005) ($M_{i,j}$ is $Exp(1)$)

3-Dimensional Assignment Problem

Three-dimensional Axial Assignment problem.
Let M be a real $n \times n \times n$ tensor of costs.
Problem: Minimise

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} M_{i,j,k}x_{i,j,k} \qquad \text{subject to}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{i,j,k} = 1, \qquad k \in [n]$$

$$\sum_{i=1}^{n}\sum_{k=1}^{n} x_{i,j,k} = 1, \qquad j \in [n]$$

$$\sum_{j=1}^{n}\sum_{k=1}^{n} x_{i,j,k} = 1, \qquad i \in [n]$$

$$x_{i,j,k} \in \{0,1\}$$

Each solution has a unique one in every “plane” of the cube.

3-Dimensional Assignment Problem

Three-dimensional Axial Assignment problem.

Let M be a real $n \times n \times n$ tensor of costs.

Problem: Minimise

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} M_{i,j,k}x_{i,j,k} \qquad \text{subject to}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{i,j,k} = 1, \qquad k \in [n]$$

$$\sum_{i=1}^{n}\sum_{k=1}^{n} x_{i,j,k} = 1, \qquad j \in [n]$$

$$\sum_{j=1}^{n}\sum_{k=1}^{n} x_{i,j,k} = 1, \qquad i \in [n]$$

$$x_{i,j,k} \in \{0,1\}$$

Each solution has a unique one in every “plane” of the cube.

NP-hard – Karp (1972)

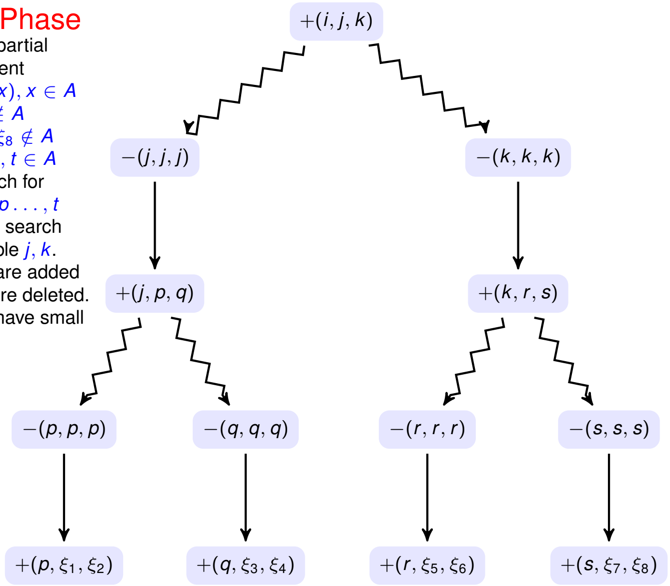
We employ a 3-phase algorithm: it has a depth parameter d.
To get a solution of value $O(n^{-(1-o(1))})$ we take $d = \epsilon \log_2 \log n$ where $\epsilon < 1/2$.

To get a feel for the algorithm, we consider $d = 2$.

Greedy Phase:
"Greedily" choose a partial assignment containing $n - n^{6/7}$ "triples" (i, j, k) at a total cost of about $n^{-6/7}$.

Main Phase

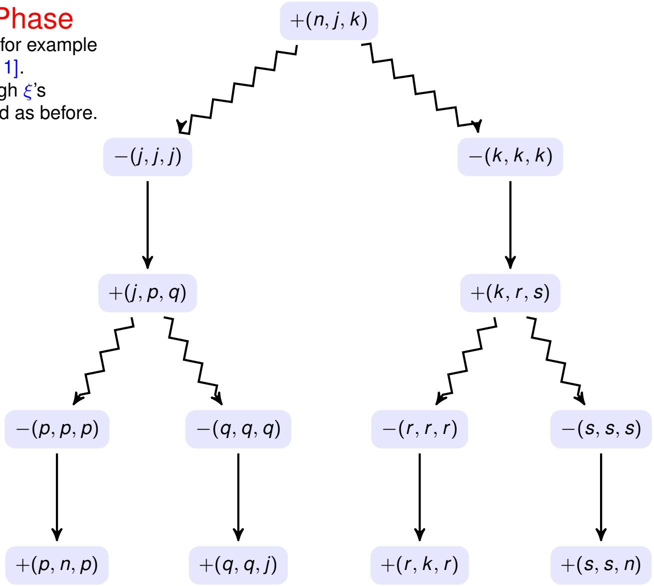
Current partial assignment is $(x, x, x), x \in A$
Here, $i \notin A$
$\xi_1, \ldots, \xi_8 \notin A$
$p, q, \ldots, t \in A$
We search for suitable $p \ldots, t$
Then we search for suitable j, k.
+triples are added
-triples are deleted.
+triples have small M-value.



3-Dimensional Assignment Problem

Final Phase

Suppose for example
$A = [n-1]$.
Not enough ξ's
to proceed as before.



3-Dimensional Assignment Problem

We have the following theorem from Frieze and Sorkin (201?):

Theorem

There is an $O(n^{3+o(1)})$ time algorithm that w.h.p. finds an assignment of value $n^{-(1-o(1))}$.

3-Dimensional Assignment Problem

We have the following theorem from Frieze and Sorkin (201?):

Theorem

There is an $O(n^{3+o(1)})$ time algorithm that w.h.p. finds an assignment of value $n^{-(1-o(1))}$.

This raises the question of what is the real growth rate of the optimum value.

One simple consequence of the breakthrough paper of Johannson, Kahn, Vu (2008) is that w.h.p. there is an assignment of value $O(\log n/n)$.

Contents of talk

(a) Random Discrete Structures

(b) Random Instances of the TSP in the unit square $[0, 1]^2$

(c) The Random Graphs $G_{n,m}$ and $G_{n,p}$.

  (1) Evolution
  (2) Chromatic number
  (3) Matchings
  (4) Hamilton cycles

(d) Randomly edge weighted graphs

  1. Minimum Spanning Tree
  2. Shortest Paths
  3. 3-Dimensional Assignment Problem
  4. Random Instances of the TSP with independent costs

(e) Random k-SAT

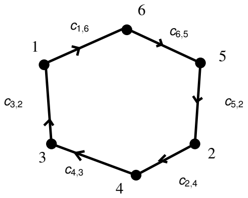
(f) Open Problems

TSP with independent costs:

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

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



$$\pi(1) = 6,\ \pi(2) = 4$$
$$\pi(3) = 1,\ \pi(4) = 3$$
$$\pi(5) = 2,\ \pi(6) = 5$$

TSP with independent costs:

Assignment problem The aim is to compute

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

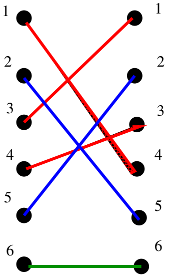
$$\frac{\pi^2}{6} \sim A(C) \leq T(C) \leq A(C) + o(1) \ w.h.p.$$

The LHS is due to Aldous (1992,2001); Nair,Prabhakar and Sharma (2006); Linusson and Wästlund (2004).
The RHS is due to Karp (1979).

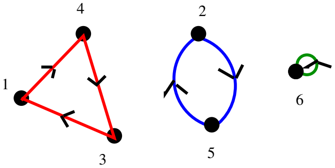
$A(C)$ is solvable in polynomial time.

TSP with independent costs:

There are two equivalent ways of viewing the assignment problem:



Minimum Weight Perfect Matching



Minimum Weight Cycle Cover

The TSP can then be thought of as finding a minimum weight cycle cover in which there is only one cycle.

TSP with independent costs:

Karp's Patching Algorithm:

TSP with independent costs:

Karp's Patching Algorithm:

- Solve the associated assignment problem.
- Patch the cycles together to get a tour.

TSP with independent costs:

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.

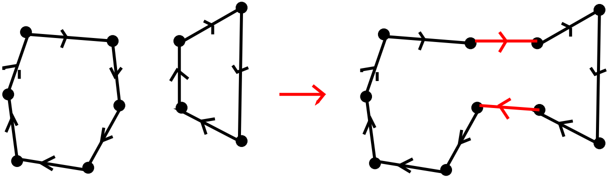


Figure: Patching two cycles

TSP with independent costs:

Theorem (Karp (1979))

W.h.p. $GAP = T(C) - A(C) = o(1)$.

TSP with independent costs:

Theorem (Karp (1979))

W.h.p. $GAP = T(C) - A(C) = o(1)$.

Theorem (Karp and Steele (1985))

W.h.p. $GAP = T(C) - A(C) = O(n^{-1/2})$.

TSP with independent costs:

Theorem (Karp (1979))

W.h.p. $GAP = T(C) - A(C) = o(1)$.

Theorem (Karp and Steele (1985))

W.h.p. $GAP = T(C) - A(C) = O(n^{-1/2})$.

By making the cycles large before doing the patching we have

Theorem (Dyer and Frieze (1990))

W.h.p. $GAP = T(C) - A(C) = o\left(\frac{\log^4 n}{n}\right)$.

TSP with independent costs:

Theorem (Karp (1979))

W.h.p. $GAP = T(C) - A(C) = o(1)$.

Theorem (Karp and Steele (1985))

W.h.p. $GAP = T(C) - A(C) = O(n^{-1/2})$.

By making the cycles large before doing the patching we have

Theorem (Dyer and Frieze (1990))

W.h.p. $GAP = T(C) - A(C) = o\left(\frac{\log^4 n}{n}\right)$.

With more care

Theorem (Frieze and Sorkin (2007))

W.h.p. $GAP = T(C) - A(C) = O\left(\frac{\log^2 n}{n}\right)$.

TSP with independent costs:

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 $n_0 = n \log \log n / \log n$.

TSP with independent costs:

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 $o(\log^2 n/n)$.

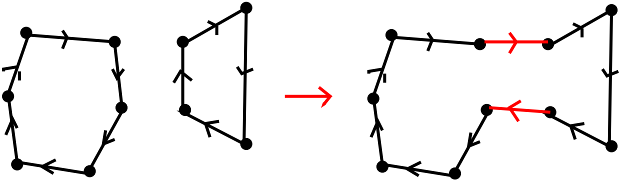


Figure: Patching two cycles

The probability we cannot patch a pair of cycles is at most

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

TSP with independent costs:

Increasing the cycle size:

- Partition the edges into
  red edges $E_1 = \{(i,j) : c_{i,j} \leq L = K \log n\}$,
  blue edges $E_2 = \{(i,j) : c_{i,j} \in [L, 2L]\}$,
  green edges $E_3 = \{(i,j) : c_{i,j} \in [2L, 3L]\}$ and
  uncolored edges $E_4 = [n]^2 / (E_1 \cup E_2 \cup E_3)$.

TSP with independent costs:

Increasing the cycle size:

- Partition the edges into
  red edges $E_1 = \{(i,j) : c_{i,j} \leq L = K \log n\}$,
  blue edges $E_2 = \{(i,j) : c_{i,j} \in [L, 2L]\}$,
  green edges $E_3 = \{(i,j) : c_{i,j} \in [2L, 3L]\}$ and
  uncolored edges $E_4 = [n]^2 / (E_1 \cup E_2 \cup E_3)$.

- Compute the optimal assignment only using edges E_1.

TSP with independent costs:

Increasing the cycle size:

- Partition the edges into
  red edges $E_1 = \{(i,j) : c_{i,j} \leq L = K \log n\}$,
  blue edges $E_2 = \{(i,j) : c_{i,j} \in [L, 2L]\}$,
  green edges $E_3 = \{(i,j) : c_{i,j} \in [2L, 3L]\}$ and
  uncolored edges $E_4 = [n]^2 / (E_1 \cup E_2 \cup E_3)$.

- Compute the optimal assignment only using edges E_1.

- Only using the edges in $E_1 \cup E_2$, increase the minimum cycle size to $n_0 = n \log \log n / \log n$.

TSP with independent costs:

Increasing the cycle size:

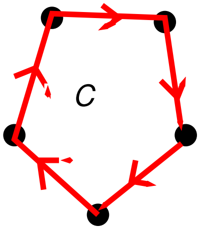
- Partition the edges into
  red edges $E_1 = \{(i,j) : c_{i,j} \leq L = K \log n\}$,
  blue edges $E_2 = \{(i,j) : c_{i,j} \in [L, 2L]\}$,
  green edges $E_3 = \{(i,j) : c_{i,j} \in [2L, 3L]\}$ and
  uncolored edges $E_4 = [n]^2 / (E_1 \cup E_2 \cup E_3)$.

- Compute the optimal assignment only using edges E_1.

- Only using the edges in $E_1 \cup E_2$, increase the minimum cycle size to $n_0 = n \log \log n / \log n$.

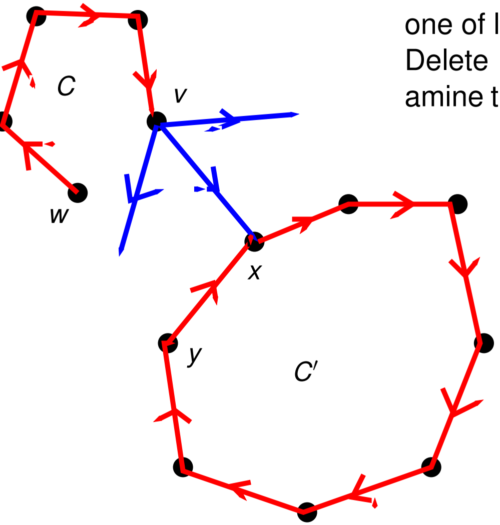
- Using the edges in E_3 only, patch the cycles into a tour.

TSP with independent costs:

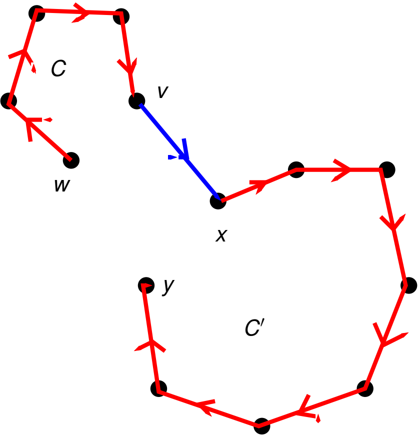


Choose some small cycle C i.e. one of length less than n_0.

TSP with independent costs:

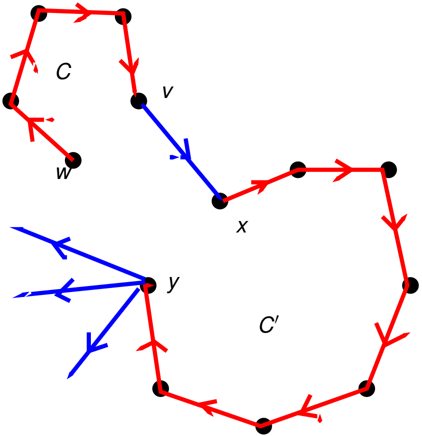


Choose some small cycle C i.e. one of length less than n_0. Delete an edge (v, w) and examine the blue edges with tail v.



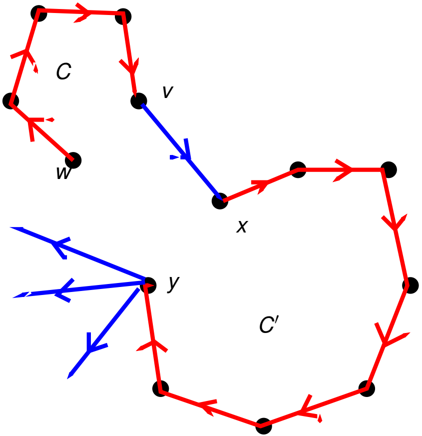
Choose some small cycle C i.e. one of length less than n_0.
Delete an edge (v, w) and examine the blue edges with tail v.
Suppose that we have blue edge (v, x) with x in a large cycle C'.

TSP with independent costs:



Choose some small cycle C i.e. one of length less than n_0.
Delete an edge (v, w) and examine the blue edges with tail v.
Suppose that we have blue edge (v, x) with x in a large cycle C'.
Delete the edge (y, x) of C'.

TSP with independent costs:

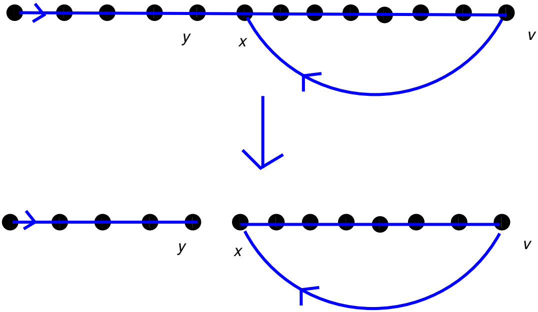


Choose some small cycle C i.e. one of length less than n_0.
Delete an edge (v, w) and examine the blue edges with tail v.
Suppose that we have blue edge (v, x) with x in a large cycle C'.
Delete the edge (y, x) of C'.

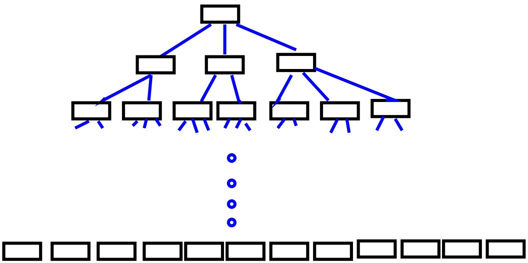
By repeating this operation, which we call a splice, we create a large number of partitions of $[n]$ into a path with initial vertex w and a collection of cycles. We call such a collection a Near

TSP with independent costs:

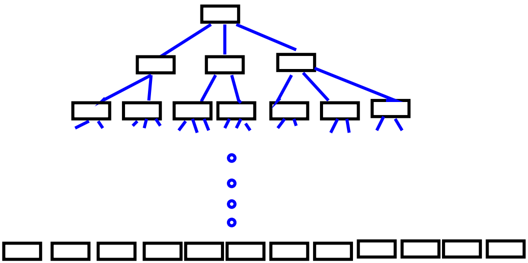
Another possibility for a splice:



In the second version of the splice we insist that the resultant path and cycle, both have length at least n_0.



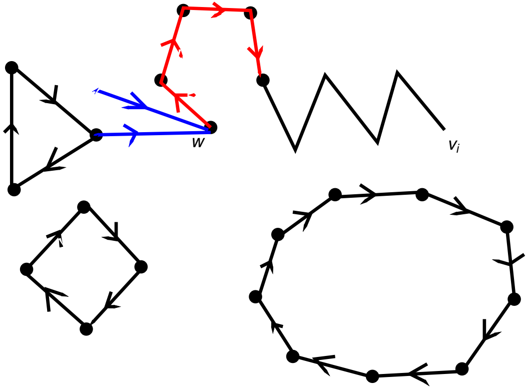
In the above diagram, each rectangle is an NPD that is obtained from its parent by a splice. A node ν is allowed d_ν children. Fot the root ρ we have $d_\rho = \Theta(\log n)$ for any other node ν we have $d_\nu = \Theta(1)$. We use the cheapest available edges to extend our path. If we build this tree to depth $\sim \log n/2$ then at the bottom of the tree there will be $n^{1/2+o(1)}$ leaves.



In the above diagram, each rectangle is an NPD that is obtained from its parent by a splice. A node ν is allowed d_ν children. Fot the root ρ we have $d_\rho = \Theta(\log n)$ for any other node ν we have $d_\nu = \Theta(1)$. We use the cheapest available edges to extend our path. If we build this tree to depth $\sim \log n/2$ then at the bottom of the tree there will be $n^{1/2+o(1)}$ leaves.

We can assume that the ith leaf corresponds to an NPD where

Now for each v_i we build a tree of NPD's where we begin by examining the edges into w.



TSP with independent costs:

Let the leaves of the ith tree have paths from $w_{i,j}$ to v_i for $j = 1, 2, \ldots, n^{1/2+o(1)}$.

TSP with independent costs:

Let the leaves of the ith tree have paths from $w_{i,j}$ to v_i for $j = 1, 2, \dots, n^{1/2+o(1)}$.

- An edge v_i to $w_{i,j}$ results in a cycle cover with (at least) one less small cycle.

TSP with independent costs:

Let the leaves of the ith tree have paths from $w_{i,j}$ to v_i for $j = 1, 2, \ldots, n^{1/2+o(1)}$.

- An edge v_i to $w_{i,j}$ results in a cycle cover with (at least) one less small cycle.

- The probability there is no such edge is at most

$$\left(1 - O\left(\frac{\log n}{n}\right)\right)^{n^{1+o(1)}} = o\left(\frac{1}{\log n}\right)$$

TSP with independent costs:

Let the leaves of the ith tree have paths from $w_{i,j}$ to v_i for $j = 1, 2, \ldots, n^{1/2+o(1)}$.

- An edge v_i to $w_{i,j}$ results in a cycle cover with (at least) one less small cycle.

- The probability there is no such edge is at most

$$\left(1 - O\left(\frac{\log n}{n}\right)\right)^{n^{1+o(1)}} = o\left(\frac{1}{\log n}\right)$$

- The cost of removing all small cycles is evidently

$$O(\log n) \times O(\log n) \times O\left(\frac{\log n}{n}\right) = O\left(\frac{\log^3 n}{n}\right).$$

TSP with independent costs:

Let the leaves of the ith tree have paths from $w_{i,j}$ to v_i for $j = 1, 2, \ldots, n^{1/2+o(1)}$.

- An edge v_i to $w_{i,j}$ results in a cycle cover with (at least) one less small cycle.
- The cost of removing all small cycles is evidently

$$O(\log n) \times O(\log n) \times O\left(\frac{\log n}{n}\right) = O\left(\frac{\log^3 n}{n}\right).$$

- The factor $O(\log n) \times O\left(\frac{\log n}{n}\right)$ can be replaced by $O\left(\frac{\log n}{n}\right)$ with some care.

TSP with independent costs:

Theorem (Held and Karp (1962))

In the worst-case the TSP can be solved exactly in time $O(n^2 2^n)$.

TSP with independent costs:

Theorem (Held and Karp (1962))

In the worst-case the TSP can be solved exactly in time $O(n^2 2^n)$.

Theorem (Frieze and Sorkin (2007))

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

TSP with independent costs:

Let

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

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 $\leq c_1 \frac{(\log n)^2}{n^{3/2}}$.

Let

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

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 at most

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

sets.

TSP with independent costs:

It is natural to consider the case where we contrain $c_{i,j} = c_{j,i}$ i.e consider a weighted graph rather than digraph.

TSP with independent costs:

It is natural to consider the case where we contrain $c_{i,j} = c_{j,i}$ i.e consider a weighted graph rather than digraph.

It is natural to replace the assignment problem by that of finding a minimum weight 2-factor viz. a collection of vertex disjoint cycles that cover every vertex.

TSP with independent costs:

It is natural to consider the case where we contrain $c_{i,j} = c_{j,i}$ i.e consider a weighted graph rather than digraph.

It is natural to replace the assignment problem by that of finding a minimum weight 2-factor viz. a collection of vertex disjoint cycles that cover every vertex.

We lose control over the number of cycles in the minimum weight 2-factor. In the following theorem we only had a bound of $O(n/\log n)$ for this.

Theorem (Frieze (2004))

W.h.p. $T(C) - 2FAC(C) = o(1)$.

Contents of talk

(a) Random Discrete Structures

(b) Random Instances of the TSP in the unit square $[0, 1]^2$

(c) The Random Graphs $G_{n,m}$ and $G_{n,p}$.

  (1) Evolution
  (2) Chromatic number
  (3) Matchings
  (4) Hamilton cycles

(d) Randomly edge weighted graphs

  1. Minimum Spanning Tree
  2. Shortest Paths
  3. 3-Dimensional Assignment Problem
  4. Random Instances of the TSP with independent costs

(e) Random k-SAT

(f) Open Problems

Random k-SAT

Variables $V = \{x_1, x_2, \ldots, x_n\}$
Literals $L = \{x_1, \bar{x}_1, x_2, \bar{x}_2, \ldots, x_n, \bar{x}_n\}$
Negated variables $\bar{V} = \{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n\}$
Clause: $S \subseteq L$

Random k-SAT

Variables $V = \{x_1, x_2, \ldots, x_n\}$
Literals $L = \{x_1, \bar{x}_1, x_2, \bar{x}_2, \ldots, x_n, \bar{x}_n\}$
Negated variables $\bar{V} = \{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n\}$
Clause: $S \subseteq L$

Instance I of k-SAT: Clauses $C_1, C_2, \ldots, C_m$ where $|C_i| = k$, $i = 1, 2, \ldots, m$.

Variables $V = \{x_1, x_2, \ldots, x_n\}$
Literals $L = \{x_1, \bar{x}_1, x_2, \bar{x}_2, \ldots, x_n, \bar{x}_n\}$
Negated variables $\bar{V} = \{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n\}$
Clause: $S \subseteq L$

Instance I of k-SAT: Clauses $C_1, C_2, \ldots, C_m$ where $|C_i| = k$, $i = 1, 2, \ldots, m$.

Truth Assignment $\phi : L \to \{0, 1\}$ 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 $i = 1, 2, \ldots, m$.

Random k-SAT

Variables $V = \{x_1, x_2, \dots, x_n\}$
Literals $L = \{x_1, \bar{x}_1, x_2, \bar{x}_2, \dots, x_n, \bar{x}_n\}$
Negated variables $\bar{V} = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\}$
Clause: $S \subseteq L$

Instance I of k-SAT: Clauses $C_1, C_2, \dots, C_m$ where $|C_i| = k$, $i = 1, 2, \dots, m$.

Truth Assignment $\phi : L \to \{0, 1\}$ such that $\phi(\bar{x}_j) = 1 - \phi(x_j)$ for $j = 1, 2, \dots, n$.
ϕ satisfies I if $1 \in \phi(C_i)$ for $i = 1, 2, \dots, m$.

For example $\phi(x_1) = 0$, $\phi(x_2) = \phi(x_3) = 1$ satisfies $\{\bar{x}_1, \bar{x}_2, \bar{x}_3\}$, $\{x_1, x_2, \bar{x}_3\}$, $\{\bar{x}_1, \bar{x}_2, x_3\}$.

Random k-SAT

Variables $V = \{x_1, x_2, \ldots, x_n\}$
Literals $L = \{x_1, \bar{x}_1, x_2, \bar{x}_2, \ldots, x_n, \bar{x}_n\}$
Negated variables $\bar{V} = \{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n\}$
Clause: $S \subseteq L$

Instance I of k-SAT: Clauses $C_1, C_2, \ldots, C_m$ where $|C_i| = k$, $i = 1, 2, \ldots, m$.

Truth Assignment $\phi : L \to \{0, 1\}$ 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 $i = 1, 2, \ldots, m$.

$\{\bar{x}_1, \bar{x}_2\}, \{x_1, \bar{x}_2\}, \{\bar{x}_1, x_2\}, \{\bar{x}_1, \bar{x}_2\}$ is unsatisfiable.

Random k-SAT

Variables $V = \{x_1, x_2, \ldots, x_n\}$
Literals $L = \{x_1, \bar{x}_1, x_2, \bar{x}_2, \ldots, x_n, \bar{x}_n\}$
Negated variables $\bar{V} = \{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n\}$
Clause: $S \subseteq L$

Instance I of k-SAT: Clauses $C_1, C_2, \ldots, C_m$ where $|C_i| = k$, $i = 1, 2, \ldots, m$.

Truth Assignment $\phi : L \to \{0, 1\}$ 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 $i = 1, 2, \ldots, m$.

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

Random k-SAT

Random instance I: Choose literals $\ell_1, \ell_2, \ldots, \ell_k$ independently and uniformly for each C_i.

Random k-SAT

Random instance I: Choose literals $\ell_1, \ell_2, \ldots, \ell_k$ independently and uniformly for each C_i.

Let $m = cn$. Then for a fixed ϕ,

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

Random instance I: Choose literals $\ell_1, \ell_2, \ldots, \ell_k$ independently and uniformly for each C_i.

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,

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

Random instance I: Choose literals $\ell_1, \ell_2, \ldots, \ell_k$ independently and uniformly for each C_i.

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 $c > 2^k \log 2$.

Random k-SAT

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 (1999) has come close to proving this.

Conjecture is true for $k = 2$. It is known that $c_2 = 1$.

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 (1999) has come close to proving this.

Conjecture is true for $k = 2$. It is known that $c_2 = 1$.

Now if $m = cn$ and

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

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

Random k-SAT

With Z equal to the number of satisfying assignments, the second moment method fails.

Achlioptas and Peres (2004) 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.

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

Theorem

If

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

then I is satisfiable w.h.p.

Z_1 reduces the weight of satisfying assignments with an "excess" of true literals.

Random k-SAT

Using a more complicated random variable, based on insights from Physicists, and doing more conditioning, but still using the second moment method,

Theorem (Coja-Oghlan and Panagiotou (2012))

If

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

then I is satisfiable w.h.p.

Random k-SAT

Using a more complicated random variable, based on insights from Physicists, and doing more conditioning, but still using the second moment method,

Theorem (Coja-Oghlan and Panagiotou (2012))

If

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

then I is satisfiable w.h.p.

Theorem (Coja-Oghlan (2013))

If

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

then I is satisfiable w.h.p.

Greedy Algorithms

Start with no values assigned to the variables.

Greedy Algorithms

Start with no values assigned to the variables.
Repeatedly, choose a random clause and assign a value to a variable to satisfy it.

Greedy Algorithms

Start with no values assigned to the variables.
Repeatedly, choose a random clause and assign a value to a variable to satisfy it.
Number of variables in problem goes down by one.

Greedy Algorithms

Start with no values assigned to the variables.
Repeatedly, choose a random clause and assign a value to a variable to satisfy it.
Number of variables in problem goes down by one.
Some clauses get satisfied and disappear from the problem, others shrink in size by one.

Greedy Algorithms

Start with no values assigned to the variables.
Repeatedly, choose a random clause and assign a value to a variable to satisfy it.
Number of variables in 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.

Greedy Algorithms

Start with no values assigned to the variables.
Repeatedly, choose a random clause and assign a value to a variable to satisfy it.
Number of variables in 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.
Repeat until all clauses are satisfied (success) or there is a clause remaining that is empty (fail).

Greedy Algorithms

Start with no values assigned to the variables.
Repeatedly, choose a random clause and assign a value to a variable to satisfy it.
Number of variables in 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.
Repeat until all clauses are satisfied (success) or there is a clause remaining that is empty (fail).

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

Greedy Algorithms

Start with no values assigned to the variables.
Repeatedly, choose a random clause and assign a value to a variable to satisfy it.
Number of variables in 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.
Repeat until all clauses are satisfied (success) or there is a clause remaining that is empty (fail).

Most of these 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 (2009) which finds a satisfying assignment w.h.p. provided there are at most $\frac{(1-\epsilon)2^k \log k}{k}n$ clauses.

Walksat

Start with the "all true" assignment: $\phi(x_j) = 1, \forall j$

Repeat

Choose an unsatisfied clause C

Choose a random variable from C and change its assigned value

Until instance is satisfied.

Walksat
Start with the "all true" assignment: $\phi(x_j) = 1, \forall j$
Repeat
    Choose an unsatisfied clause C
    Choose a random variable from C and change its assigned value
Until instance is satisfied.

Aleknovich and Ben-Sasson (2007) show that Walksat solves a random instance of 3-SAT w.h.p. for $m < 1.67n$.

Random k-SAT

Walksat
Start with the "all true" assignment: $\phi(x_j) = 1, \forall j$
Repeat
Choose an unsatisfied clause C
Choose a random variable from C and change its assigned value
Until instance is satisfied.

Aleknovich and Ben-Sasson (2007) show that Walksat solves a random instance of 3-SAT w.h.p. for $m < 1.67n$.

Coja-Oghlan, Feige, Frieze, Krivelevich and Vilenchik (2009) show that for large k, Walksat solves a random instance of k-SAT w.h.p. for $m/n \leq c2^k/k^2$.

Walksat

Start with the "all true" assignment: $\phi(x_j) = 1, \forall j$

Repeat

Choose an unsatisfied clause C

Choose a random variable from C and change its assigned value

Until instance is satisfied.

Aleknovich and Ben-Sasson (2007) show that Walksat solves a random instance of 3-SAT w.h.p. for $m < 1.67n$.

Coja-Oghlan, Feige, Frieze, Krivelevich and Vilenchik (2009) show that for large k, Walksat solves a random instance of k-SAT w.h.p. for $m/n \leq c2^k/k^2$.

Coja-Oghlan and Frieze (2012) show that for large k, Walksat solves a random instance of k-SAT w.h.p. for $m/n \leq c2^k/k$.

Outline of Walksat for $m/n \leq c2^k/k^2$.

Outline of Walksat for $m/n \leq c2^k/k^2$.

We say that a clause C is infected by Walksat if its assigned value could be changed in the course of the algorithm.

Let A denote the set of infected clauses and let $V_A = \bigcup_{C \in A} C$.

Outline of Walksat for $m/n \leq c2^k/k^2$.

We say that a clause C is infected by Walksat if its assigned value could be changed in the course of the algorithm.

Let A denote the set of infected clauses and let $V_A = \bigcup_{C \in A} C$.

W1 $C \subseteq \bar{V}$ implies $C \in A$.

Outline of Walksat for $m/n \leq c2^k/k^2$.
We say that a clause C is infected by Walksat if its assigned value could be changed in the course of the algorithm.

Let A denote the set of infected clauses and let $V_A = \bigcup_{C \in A} C$.

W1 $C \subseteq \bar{V}$ implies $C \in A$.

W2 If $C \cap V \subseteq V_A$ then $C \in A$

Outline of Walksat for $m/n \leq c2^k/k^2$.
We say that a clause C is infected by Walksat if its assigned value could be changed in the course of the algorithm.

Let A denote the set of infected clauses and let $V_A = \bigcup_{C \in A} C$.

W1 $C \subseteq \bar{V}$ implies $C \in A$.

W2 If $C \cap V \subseteq V_A$ then $C \in A$

Because m is small, this means that w.h.p. A is small and then $C, C' \in A$ are almost disjoint.

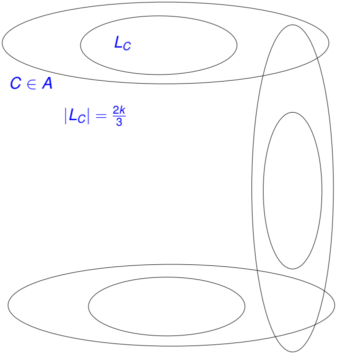


Figure: Each $C \in A$ has its own unique set of $2k/3$ literals

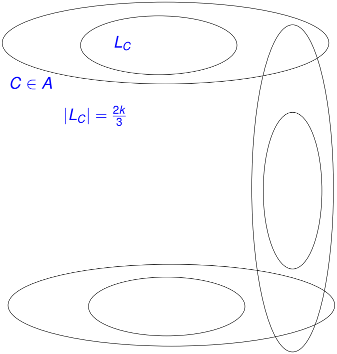


Figure: Each $C \in A$ has its own unique set of $2k/3$ literals

Putting $\sigma_A(x) = 0$ for $\bar{x} \in \bar{V} \cap \bigcup_{C \in A} L_C$ and $\sigma_A(x) = 1$ otherwise, yields a satisfying assignment.

Now consider the Hamming distance between the current assignment σ_W of Walksat and σ_A.

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).

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).

Similar idea to that of Papadimitriou (1994) for 2-SAT.

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 (2008) proved

Theorem

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

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

Contents of talk

(a) Random Discrete Structures

(b) Random Instances of the TSP in the unit square $[0, 1]^2$

(c) The Random Graphs $G_{n,m}$ and $G_{n,p}$.

  (1) Evolution
  (2) Chromatic number
  (3) Matchings
  (4) Hamilton cycles

(d) Randomly edge weighted graphs

  1. Minimum Spanning Tree
  2. Shortest Paths
  3. 3-Dimensional Assignment Problem
  4. Random Instances of the TSP with independent costs

(e) Random k-SAT

(f) Open Problems

Open Questions/Problems

Find a polynomial time algorithm that w.h.p. finds a clique of size at least $1.001 \log_2 n$ in $G_{n,1/2}$.

Open Questions/Problems

Find a polynomial time algorithm that w.h.p. finds a planted clique of size $o(n^{1/2})$ in $G_{n,1/2}$.

Find a polynomial time algorithm that w.h.p. finds a planted clique of size $o(n^{1/2})$ in $G_{n,1/2}$.

Choose a p-subset $S \subseteq [n]$ and add all edges contained in S to $G_{n,1/2}$. Now ask someone else to find S.

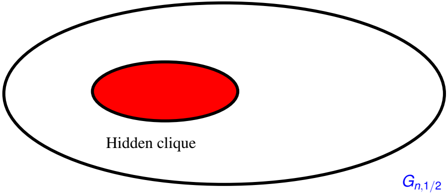


Figure: Planted Clique

Open Questions/Problems

Find a polynomial time algorithm that w.h.p. finds a planted clique of size $o(n^{1/2})$ in $G_{n,1/2}$.

Choose a p-subset $S \subseteq [n]$ and add all edges contained in S to $G_{n,1/2}$. Now ask someone else to find S.

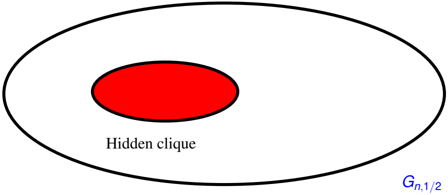


Figure: Planted Clique

If $p \gg n^{1/2}$ then enough to check vertices of high degree, Kucera (1995).

Find a polynomial time algorithm that w.h.p. finds a planted clique of size $o(n^{1/2})$ in $G_{n,1/2}$.

Choose a p-subset $S \subseteq [n]$ and add all edges contained in S to $G_{n,1/2}$. Now ask someone else to find S.

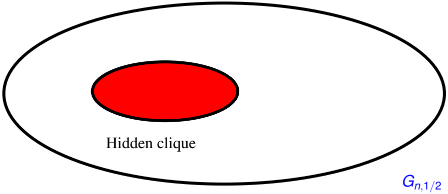


Figure: Planted Clique

If $p = O(n^{1/2})$ then spectral methods work, Alon, Krivelevich and Sudakov (1998).

Open Questions/Problems

Find a polynomial time algorithm that w.h.p. finds a planted clique of size $o(n^{1/2})$ in $G_{n,1/2}$.

Choose a p-subset $S \subseteq [n]$ and add all edges contained in S to $G_{n,1/2}$. Now ask someone else to find S.

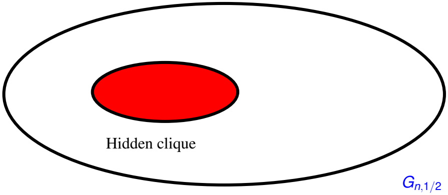
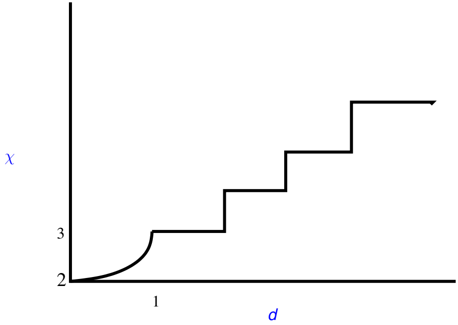


Figure: Planted Clique

If $p = o(n^{1/2})$ then there are negative results on statistical algorithms, Feldman, Grigorescu, Reyzin, Vempala and Xiao (2013).

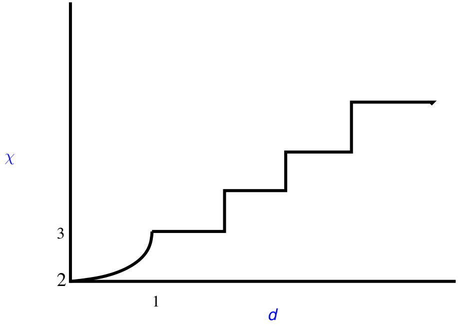
Open Questions/Problems

Find the precise threshold for the k-colorability of the random graph $G_{n,p}$.



Open Questions/Problems

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.

Open Questions/Problems

Find the precise threshold for the satisfiability of random k-SAT.

Open Questions/Problems

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.

Open Questions/Problems

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

Open Questions/Problems

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

Construct a linear time algorithm for finding a Hamilton cycle in this model.

Open Questions/Problems

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.

Open Questions/Problems

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.

In practise, branch and bound works well on these instances.

Open Questions/Problems

Analyse the *ordinary* simplex algorithm on random instances.

Open Questions/Problems

Analyse the *ordinary* simplex algorithm on random instances.

Significant results are limited to more sophisticated versions such as the shadow simplex algorithm, Borgwardt (1980).

Open Questions/Problems

Analyse the *ordinary* simplex algorithm on random instances.

Significant results are limited to more sophisticated versions such as the shadow simplex algorithm, Borgwardt (1980).

Led to the notion of smoothed analysis: Spielman and Teng (2004).

Open Questions/Problems

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.

$$\begin{vmatrix}
0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0
\end{vmatrix}$$

Open Questions/Problems

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.

Open Questions/Problems

Determine the constant β in the Beardwood, Halton and Hammersley theorem.

Open Questions/Problems

Determine the asymptotics for the value of a random multi-dimensional assignment problem and find asymptotically optimal heuristics.

Open Questions/Problems

Determine the asymptotics for the value of a random multi-dimensional assignment problem and find asymptotically optimal heuristics.

Give a uniform $[0,1]$ weight X_e to each edge of the complete 3-uniform hypergraph $H_{n:3}$. Let Z_n denote the minimum weight of a perfect matching.

Open Questions/Problems

Determine the asymptotics for the value of a random multi-dimensional assignment problem and find asymptotically optimal heuristics.

Give a uniform $[0, 1]$ weight X_e to each edge of the complete 3-uniform hypergraph $H_{n:3}$. Let Z_n denote the minimum weight of a perfect matching.

It is known that w.h.p.

$$\frac{c_1}{n} \leq Z_n \leq \frac{c_2 \log n}{n}.$$

The LHS is easy. The RHS depends on a deep result of Johansson, Kahn and Vu (2008).

Open Questions/Problems

Determine the threshold for a random subgraph of the *n*-cube to be Hamiltonian.

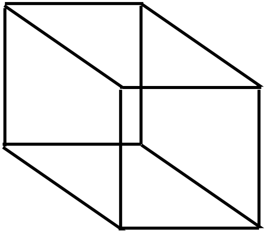


Figure: 3-cube

THANK YOU

정말 감사합니다