Part I

Probabilistic Method

WHICH IS THE MOST BEAUTIFUL EQUATION?

WHICH IS THE MOST BEAUTIFUL EQUATION?

$$e^{i\pi} + 1 = 0$$

How to prove that some object exists?

HOW TO PROVE THAT

SOME OBJECT EXISTS?

How to prove that some object exists?

HOW TO PROVE THAT

SOME OBJECT EXISTS?

- To develop a constructive method (and to show its correctness) how to find or design such an object - a constructive approach

How to prove that some object exists?

HOW TO PROVE THAT

SOME OBJECT EXISTS?

- To develop a constructive method (and to show its correctness) how to find or design such an object - a constructive approach
- To prove that probability that such an object exists is positive - a non constructive approach.

Chapter 8. PROBABILISTIC METHOD

Chapter 8. PROBABILISTIC METHOD

The probabilistic method is a powerful tool **to demonstrate the existence of some combinatorial objects**.

Chapter 8. PROBABILISTIC METHOD

The probabilistic method is a powerful tool **to demonstrate the existence of some combinatorial objects**.

In some cases this method can be used also to derive an algorithm for finding such an object.

Chapter 8. PROBABILISTIC METHOD

The probabilistic method is a powerful tool **to demonstrate the existence of some combinatorial objects**.

In some cases this method can be used also to derive an algorithm for finding such an object.

Two basic approaches of the probabilistic method:

1. **The expectation approach:** Any random variable V assumes at least one value that is not smaller than its expectation EV, and at least one value that is not greater than its expectation EV.

Chapter 8. PROBABILISTIC METHOD

The probabilistic method is a powerful tool **to demonstrate the existence of some combinatorial objects**.

In some cases this method can be used also to derive an algorithm for finding such an object.

Two basic approaches of the probabilistic method:

1. **The expectation approach:** Any random variable V assumes at least one value that is not smaller than its expectation EV, and at least one value that is not greater than its expectation EV.
2. **The sampling approach:** If an object chosen randomly from a universe/set U satisfies a property P with a positive probability, then there must be an object in U that satisfies the property P.

Chapter 8. PROBABILISTIC METHOD

The probabilistic method is a powerful tool **to demonstrate the existence of some combinatorial objects**.

In some cases this method can be used also to derive an algorithm for finding such an object.

Two basic approaches of the probabilistic method:

1. **The expectation approach:** Any random variable V assumes at least one value that is not smaller than its expectation EV, and at least one value that is not greater than its expectation EV.
2. **The sampling approach:** If an object chosen randomly from a universe/set U satisfies a property P with a positive probability, then there must be an object in U that satisfies the property P.

The above two simple ideas have a surprising power.

Chapter 8. PROBABILISTIC METHOD

The probabilistic method is a powerful tool **to demonstrate the existence of some combinatorial objects**.

In some cases this method can be used also to derive an algorithm for finding such an object.

Two basic approaches of the probabilistic method:

1. **The expectation approach:** Any random variable V assumes at least one value that is not smaller than its expectation EV, and at least one value that is not greater than its expectation EV.
2. **The sampling approach:** If an object chosen randomly from a universe/set U satisfies a property P with a positive probability, then there must be an object in U that satisfies the property P.

The above two simple ideas have a surprising power.

Their power comes from our ability to reformulate, in various ways, so called **counting arguments** in the language of probability and then *to apply* various tools of the probability theory.

EXAMPLE

Example: One can show that for every $n \times n$ 0-1-matrix A, and for any randomly chosen vector $b \in \{-1, +1\}^n$, it holds

$$\|Ab\| \leq 4\sqrt{n \ln n}$$

with probability at least $1 - \frac{2}{n}$.

EXAMPLE

Example: One can show that for every $n \times n$ 0-1-matrix A, and for any randomly chosen vector $b \in \{-1, +1\}^n$, it holds

$$\|Ab\| \leq 4\sqrt{n \ln n}$$

with probability at least $1 - \frac{2}{n}$.

From that we may conclude that for every such a matrix A, there always exists a vector $b \in \{-1, +1\}^n$ such that

$$\|Ab\| \leq 4\sqrt{n \ln n}.$$

LIMITS of PROBABILISTIC METHODS

LIMITS of PROBABILISTIC METHODS

- Probabilistic method is especially useful in case we can show that the probability is quite large that the object we look for exists and we can quite easily verify whether the random process we will create indeed found such an object.

LIMITS of PROBABILISTIC METHODS

- Probabilistic method is especially useful in case we can show that the probability is quite large that the object we look for exists and we can quite easily verify whether the random process we will create indeed found such an object.
- If such a probability is indeed large then we can find such an object quite efficiently just by applying a random searching process - a sampling experiment.

LIMITS of PROBABILISTIC METHODS

- Probabilistic method is especially useful in case we can show that the probability is quite large that the object we look for exists and we can quite easily verify whether the random process we will create indeed found such an object.
- If such a probability is indeed large then we can find such an object quite efficiently just by applying a random searching process - a sampling experiment.
- In some cases, however, no explicit construction of a combinatorial object is known in spite of the fact that we can show that such object exists.

EXAMPLE

EXAMPLE

- **Example** Using probabilistic method it can be shown that for any n there exists a sorting network that sorts n integers in parallel in $O(\log n)$ time.

EXAMPLE

- **Example** Using probabilistic method it can be shown that for any n there exists a sorting network that sorts n integers in parallel in $O(\log n)$ time.
- A method is known to construct for any n a sorting network that sorts n integers in parallel in $O(\log^2 n)$ time.

EXAMPLE

- **Example** Using probabilistic method it can be shown that for any n there exists a sorting network that sorts n integers in parallel in $O(\log n)$ time.
- A method is known to construct for any n a sorting network that sorts n integers in parallel in $O(\log^2 n)$ time.
- No method is known to construct for any n a sorting network that could sort n integers in parallel in $O(\log n)$ time.

EXAMPLE

- **Example** Using probabilistic method it can be shown that for any n there exists a sorting network that sorts n integers in parallel in $O(\log n)$ time.
- A method is known to construct for any n a sorting network that sorts n integers in parallel in $O(\log^2 n)$ time.
- No method is known to construct for any n a sorting network that could sort n integers in parallel in $O(\log n)$ time. In spite of many efforts to do that during the last 25 years.

BASIC IDEA and an EXAMPLE

Probabilistic method consists of two stages.

BASIC IDEA and an EXAMPLE

Probabilistic method consists of two stages.

1. A "thought experiment" $\mathcal{E}$ is designed in which a carefully chosen random process (called usually as *an experiment* - for example, a dice tossing) P plays a key role.

BASIC IDEA and an EXAMPLE

Probabilistic method consists of two stages.

1. A "thought experiment" $\mathcal{E}$ is designed in which a carefully chosen random process (called usually as *an experiment* - for example, a dice tossing) P plays a key role.
2. The random process P is then analyzed and some conclusions are made that are, or at least look as, independent of the experiment $\mathcal{E}$.

EXAMPLE - MAX-CUT PROBLEM

EXAMPLE - MAX-CUT PROBLEM

Problem: Given is an undirected graph $G = (V, E)$ with $n = |V|$ vertices and $m = |E|$ edges.

EXAMPLE - MAX-CUT PROBLEM

Problem: Given is an undirected graph $G = (V, E)$ with $n = |V|$ vertices and $m = |E|$ edges. The task is to partition vertices in V into two sets A and B in such a way that maximizes the number of such edges (u, v), where $u \in A, v \in B$.

EXAMPLE - MAX-CUT PROBLEM

Problem: Given is an undirected graph $G = (V, E)$ with $n = |V|$ vertices and $m = |E|$ edges. The task is to partition vertices in V into two sets A and B in such a way that maximizes the number of such edges (u, v), where $u \in A, v \in B$.

Theorem: For any undirected graph $G = (V, E)$ with n vertices and m edges, there is a partition of the vertex set V into two sets A and B such that

$$|\{(u, v) \in E \mid u \in A, v \in B\}| \geq \frac{m}{2}.$$

EXAMPLE - MAX-CUT PROBLEM

Problem: Given is an undirected graph $G = (V, E)$ with $n = |V|$ vertices and $m = |E|$ edges. The task is to partition vertices in V into two sets A and B in such a way that maximizes the number of such edges (u, v), where $u \in A, v \in B$.

Theorem: For any undirected graph $G = (V, E)$ with n vertices and m edges, there is a partition of the vertex set V into two sets A and B such that

$$|\{(u, v) \in E \mid u \in A, v \in B\}| \geq \frac{m}{2}.$$

Proof: Let us consider the following experiment: Each vertex of G is independently and equiprobably assigned to either A or B.

EXAMPLE - MAX-CUT PROBLEM

Problem: Given is an undirected graph $G = (V, E)$ with $n = |V|$ vertices and $m = |E|$ edges. The task is to partition vertices in V into two sets A and B in such a way that maximizes the number of such edges (u, v), where $u \in A, v \in B$.

Theorem: For any undirected graph $G = (V, E)$ with n vertices and m edges, there is a partition of the vertex set V into two sets A and B such that

$$|\{(u, v) \in E \mid u \in A, v \in B\}| \geq \frac{m}{2}.$$

Proof: Let us consider the following experiment: Each vertex of G is independently and equiprobably assigned to either A or B.

For any edge (u, v), the probability that its end-vertices are in different sets is $\frac{1}{2}$.

EXAMPLE - MAX-CUT PROBLEM

Problem: Given is an undirected graph $G = (V, E)$ with $n = |V|$ vertices and $m = |E|$ edges. The task is to partition vertices in V into two sets A and B in such a way that maximizes the number of such edges (u, v), where $u \in A, v \in B$.

Theorem: For any undirected graph $G = (V, E)$ with n vertices and m edges, there is a partition of the vertex set V into two sets A and B such that

$$|\{(u, v) \in E \mid u \in A, v \in B\}| \geq \frac{m}{2}.$$

Proof: Let us consider the following experiment: Each vertex of G is independently and equiprobably assigned to either A or B.

For any edge (u, v), the probability that its end-vertices are in different sets is $\frac{1}{2}$.

By linearity of expectations, the expected number of edges with end-vertices in different sets is thus

$$\mathbf{E}[\text{cut-size}] = \frac{m}{2}.$$

EXAMPLE - MAX-CUT PROBLEM

Problem: Given is an undirected graph $G = (V, E)$ with $n = |V|$ vertices and $m = |E|$ edges. The task is to partition vertices in V into two sets A and B in such a way that maximizes the number of such edges (u, v), where $u \in A, v \in B$.

Theorem: For any undirected graph $G = (V, E)$ with n vertices and m edges, there is a partition of the vertex set V into two sets A and B such that

$$|\{(u, v) \in E \mid u \in A, v \in B\}| \geq \frac{m}{2}.$$

Proof: Let us consider the following experiment: Each vertex of G is independently and equiprobably assigned to either A or B.

For any edge (u, v), the probability that its end-vertices are in different sets is $\frac{1}{2}$.

By linearity of expectations, the expected number of edges with end-vertices in different sets is thus

$$\mathbf{E}[\text{cut-size}] = \frac{m}{2}.$$

That implies that there must be a partition satisfying the theorem.

DERANDOMIZATION

DERANDOMIZATION

- In some cases the proof of existence, of an object O, obtained by the probabilistic method can be converted into an efficient randomized algorithm to find O.

DERANDOMIZATION

- In some cases the proof of existence, of an object O, obtained by the probabilistic method can be converted into an efficient randomized algorithm to find O.
- In some other cases the existence proof obtained by the probabilistic method can be converted even to an efficient deterministic algorithm **to find a desirable object** O- such a process is called derandomization.

FROM THE PROOF of EXISTENCE to LAS VEGAS ALGORITHM for MAX-CUT PROBLEM

FROM THE PROOF of EXISTENCE to LAS VEGAS ALGORITHM for MAX-CUT PROBLEM

We show how to transform the argument from slide #4 about the existence of a partition with at least $\frac{m}{2}$ of edges to a Las Vegas algorithm.

FROM THE PROOF of EXISTENCE to LAS VEGAS ALGORITHM for MAX-CUT PROBLEM

We show how to transform the argument from slide #4 about the existence of a partition with at least $\frac{m}{2}$ of edges to a Las Vegas algorithm.

Let us design a partition $C(A, B)$ using randomization described above and denote

FROM THE PROOF of EXISTENCE to LAS VEGAS ALGORITHM for MAX-CUT PROBLEM

We show how to transform the argument from slide #4 about the existence of a partition with at least $\frac{m}{2}$ of edges to a Las Vegas algorithm.

Let us design a partition $C(A, B)$ using randomization described above and denote

$$p = Pr(C(A, B) < \frac{m}{2})$$

FROM THE PROOF of EXISTENCE to LAS VEGAS ALGORITHM for MAX-CUT PROBLEM

We show how to transform the argument from slide #4 about the existence of a partition with at least $\frac{m}{2}$ of edges to a Las Vegas algorithm.

Let us design a partition $C(A, B)$ using randomization described above and denote

$$p = Pr(C(A,B) < \frac{m}{2})$$

Then

$$\frac{m}{2} \quad = \quad \mathbf{E}[C(A,B)] = \sum_{i \leq m/2-1} iPr(C(A,B) = i) + \sum_{m/2 \leq i \leq m} iPr(C(A,B) = i) \geq (1-p)\left(\frac{m}{2} + 1\right)$$

FROM THE PROOF of EXISTENCE to LAS VEGAS ALGORITHM for MAX-CUT PROBLEM

We show how to transform the argument from slide #4 about the existence of a partition with at least $\frac{m}{2}$ of edges to a Las Vegas algorithm.

Let us design a partition $C(A, B)$ using randomization described above and denote

$$p = Pr(C(A, B) < \frac{m}{2})$$

Then

$$\frac{m}{2} \quad = \quad \mathbf{E}[C(A, B)] = \sum_{i \leq m/2-1} iPr(C(A, B) = i) + \sum_{m/2 \leq i \leq m} iPr(C(A, B) = i) \geq (1-p)\left(\frac{m}{2}+1\right)$$

which implies that

$$p \geq \frac{1}{m/2+1}$$

FROM THE PROOF of EXISTENCE to LAS VEGAS ALGORITHM for MAX-CUT PROBLEM

We show how to transform the argument from slide #4 about the existence of a partition with at least $\frac{m}{2}$ of edges to a Las Vegas algorithm.

Let us design a partition $C(A, B)$ using randomization described above and denote

$$p = Pr(C(A,B) < \frac{m}{2})$$

Then

$$\frac{m}{2} \quad = \quad \mathbf{E}[C(A,B)] = \sum_{i \le m/2-1} iPr(C(A,B) = i) + \sum_{m/2 \le i \le m} iPr(C(A,B) = i) \ge (1-p)\left(\frac{m}{2} + 1\right)$$

which implies that

$$p \ge \frac{1}{m/2 + 1}$$

The expected number of samples before finding a cut with value at least $\frac{m}{2}$ is therefore $\frac{m}{2} + 1$.

FROM THE PROOF of EXISTENCE to LAS VEGAS ALGORITHM for MAX-CUT PROBLEM

We show how to transform the argument from slide #4 about the existence of a partition with at least $\frac{m}{2}$ of edges to a Las Vegas algorithm.

Let us design a partition $C(A, B)$ using randomization described above and denote

$$p = Pr(C(A, B) < \frac{m}{2})$$

Then

$$\frac{m}{2} = \mathbf{E}[C(A, B)] = \sum_{i \leq m/2-1} i Pr(C(A, B) = i) + \sum_{m/2 \leq i \leq m} i Pr(C(A, B) = i) \geq (1 - p)\left(\frac{m}{2} + 1\right)$$

which implies that

$$p \geq \frac{1}{m/2 + 1}$$

The expected number of samples before finding a cut with value at least $\frac{m}{2}$ is therefore $\frac{m}{2} + 1$.

Since we can test in polynomial time whether the value of the cut determined by a sample is at least $m/2$, by counting edges crossing the cut, we have a Las Vegas algorithm to find a cut.

ASSIGNMENT of HATS - I.

ASSIGNMENT of HATS - I.

There are n robots in a field and each of them can see only k other robots - n, k are fixed.

ASSIGNMENT of HATS - I.

There are n robots in a field and each of them can see only k other robots - n, k are fixed.

Each robot wants to have a hat and a robot can be happy only if she has a hat and none of robots she sees has a hat.

ASSIGNMENT of HATS - I.

There are *n* robots in a field and each of them can see only *k* other robots - n, k are fixed.

Each robot wants to have a hat and a robot can be happy only if she has a hat and none of robots she sees has a hat.

How many robots can be happy?

ASSIGNMENT of HATS - II.

There are n robots in a field and each of them can see only k other robots for fixed n, k.

ASSIGNMENT of HATS - II.

There are *n* robots in a field and each of them can see only *k* other robots for fixed n, k. **Each robot wants to have a hat and a robot can be happy only if she has a hat and none of robots she sees has a hat.**

ASSIGNMENT of HATS - II.

There are *n* robots in a field and each of them can see only *k* other robots for fixed n, k. **Each robot wants to have a hat and a robot can be happy only if she has a hat and none of robots she sees has a hat. How many robots can be happy?**

ASSIGNMENT of HATS - II.

There are n robots in a field and each of them can see only k other robots for fixed n, k. **Each robot wants to have a hat and a robot can be happy only if she has a hat and none of robots she sees has a hat. How many robots can be happy?**

Experiment: Let us give to each robot a hat with a probability p.

ASSIGNMENT of HATS - II.

There are n robots in a field and each of them can see only k other robots for fixed n, k. **Each robot wants to have a hat and a robot can be happy only if she has a hat and none of robots she sees has a hat. How many robots can be happy?**

Experiment: Let us give to each robot a hat with a probability p. Then the probability that any particular robot is happy is $p(1-p)^k$.

ASSIGNMENT of HATS - II.

There are n robots in a field and each of them can see only k other robots for fixed n, k. **Each robot wants to have a hat and a robot can be happy only if she has a hat and none of robots she sees has a hat. How many robots can be happy?**

Experiment: Let us give to each robot a hat with a probability p. Then the probability that any particular robot is happy is $p(1-p)^k$.

If X_r is the indicator variable for the event that robot number r is happy,

ASSIGNMENT of HATS - II.

There are n robots in a field and each of them can see only k other robots for fixed n, k. **Each robot wants to have a hat and a robot can be happy only if she has a hat and none of robots she sees has a hat. How many robots can be happy?**

Experiment: Let us give to each robot a hat with a probability p. Then the probability that any particular robot is happy is $p(1-p)^k$.

If X_r is the indicator variable for the event that robot number r is happy, then $\mathbf{E}[X_r] =$

ASSIGNMENT of HATS - II.

There are n robots in a field and each of them can see only k other robots for fixed n, k. **Each robot wants to have a hat and a robot can be happy only if she has a hat and none of robots she sees has a hat. How many robots can be happy?**

Experiment: Let us give to each robot a hat with a probability p. Then the probability that any particular robot is happy is $p(1-p)^k$.

If X_r is the indicator variable for the event that robot number r is happy, then $\mathbf{E}[X_r] = p(1-p)^k$ and the expected number of happy robots is $np(1-p)^k$

ASSIGNMENT of HATS - II.

There are n robots in a field and each of them can see only k other robots for fixed n, k. **Each robot wants to have a hat and a robot can be happy only if she has a hat and none of robots she sees has a hat. How many robots can be happy?**

Experiment: Let us give to each robot a hat with a probability p. Then the probability that any particular robot is happy is $p(1-p)^k$.

If X_r is the indicator variable for the event that robot number r is happy, then $\mathbf{E}[X_r] = p(1-p)^k$ and the expected number of happy robots is $np(1-p)^k$ and so there has to be a specific assignment of hats that so many robots are happy.

ASSIGNMENT of HATS - II.

There are n robots in a field and each of them can see only k other robots for fixed n, k. **Each robot wants to have a hat and a robot can be happy only if she has a hat and none of robots she sees has a hat. How many robots can be happy?**

Experiment: Let us give to each robot a hat with a probability p. Then the probability that any particular robot is happy is $p(1-p)^k$.

If X_r is the indicator variable for the event that robot number r is happy, then $\mathbf{E}[X_r] = p(1-p)^k$ and the expected number of happy robots is $np(1-p)^k$ and so there has to be a specific assignment of hats that so many robots are happy.

Next task is to find such a p that maximizes the value $np(1-p)^k$.

ASSIGNMENT of HATS - II.

There are n robots in a field and each of them can see only k other robots for fixed n, k. **Each robot wants to have a hat and a robot can be happy only if she has a hat and none of robots she sees has a hat. How many robots can be happy?**

Experiment: Let us give to each robot a hat with a probability p. Then the probability that any particular robot is happy is $p(1-p)^k$.

If X_r is the indicator variable for the event that robot number r is happy, then $\mathbf{E}[X_r] = p(1-p)^k$ and the expected number of happy robots is $np(1-p)^k$ and so there has to be a specific assignment of hats that so many robots are happy.

Next task is to find such a p that maximizes the value $np(1-p)^k$. From the equation

$$\frac{d}{dp}(np(1-p)^k) = 0$$

ASSIGNMENT of HATS - II.

There are n robots in a field and each of them can see only k other robots for fixed n, k. **Each robot wants to have a hat and a robot can be happy only if she has a hat and none of robots she sees has a hat. How many robots can be happy?**

Experiment: Let us give to each robot a hat with a probability p. Then the probability that any particular robot is happy is $p(1-p)^k$.

If X_r is the indicator variable for the event that robot number r is happy, then $\mathbf{E}[X_r] = p(1-p)^k$ and the expected number of happy robots is $np(1-p)^k$ and so there has to be a specific assignment of hats that so many robots are happy.

Next task is to find such a p that maximizes the value $np(1-p)^k$. From the equation

$$\frac{d}{dp}(np(1-p)^k) = 0$$

we get $p = 1/(k+1)$.

ASSIGNMENT of HATS - II.

There are n robots in a field and each of them can see only k other robots for fixed n, k. **Each robot wants to have a hat and a robot can be happy only if she has a hat and none of robots she sees has a hat. How many robots can be happy?**

Experiment: Let us give to each robot a hat with a probability p. Then the probability that any particular robot is happy is $p(1-p)^k$.

If X_r is the indicator variable for the event that robot number r is happy, then $\mathbf{E}[X_r] = p(1-p)^k$ and the expected number of happy robots is $np(1-p)^k$ and so there has to be a specific assignment of hats that so many robots are happy.

Next task is to find such a p that maximizes the value $np(1-p)^k$. From the equation

$$\frac{d}{dp}(np(1-p)^k) = 0$$

we get $p = 1/(k+1)$.For this p and large k, the expected number of hats is

$$np(1-p)^k = n\left(\frac{1}{k+1}\right)\left(1 - \frac{1}{k+1}\right)^k \leq \frac{n}{(k+1)e^k}$$

HAMILTONIAN PATHS in TOURNAMENTS

A **tournament** is a complete directed graph.

HAMILTONIAN PATHS in TOURNAMENTS

A **tournament** is a complete directed graph.

A **Hamiltonian path** in a graph $G = (V, E)$ is a path that visits each vertex (representing a player) of V exactly once.

HAMILTONIAN PATHS in TOURNAMENTS

A **tournament** is a complete directed graph.

A **Hamiltonian path** in a graph $G = (V, E)$ is a path that visits each vertex (representing a player) of V exactly once.

Theorem: Every tournament has a Hamiltonian path.

HAMILTONIAN PATHS in TOURNAMENTS

A **tournament** is a complete directed graph.

A **Hamiltonian path** in a graph $G = (V, E)$ is a path that visits each vertex (representing a player) of V exactly once.

Theorem: Every tournament has a Hamiltonian path.

Proof will be by the induction on the number n of vertices in a tournament.

HAMILTONIAN PATHS in TOURNAMENTS

A **tournament** is a complete directed graph.

A **Hamiltonian path** in a graph $G = (V, E)$ is a path that visits each vertex (representing a player) of V exactly once.

Theorem: Every tournament has a Hamiltonian path.

Proof will be by the induction on the number n of vertices in a tournament.

Induction step. Suppose that every tournament with at most n vertices has a Hamiltonian path and let a tournament $T = (V, E)$ with $n + 1$ vertices be given.

HAMILTONIAN PATHS in TOURNAMENTS

A **tournament** is a complete directed graph.

A **Hamiltonian path** in a graph $G = (V, E)$ is a path that visits each vertex (representing a player) of V exactly once.

Theorem: Every tournament has a Hamiltonian path.

Proof will be by the induction on the number n of vertices in a tournament.

Induction step. Suppose that every tournament with at most n vertices has a Hamiltonian path and let a tournament $T = (V, E)$ with $n + 1$ vertices be given.

Choose any vertex v and define two sets of vertices

$$A = \{u \,|\, (u, v) \in E\} \qquad B = \{u \,|\, (v, u) \in E\}.$$

HAMILTONIAN PATHS in TOURNAMENTS

A **tournament** is a complete directed graph.

A **Hamiltonian path** in a graph $G = (V, E)$ is a path that visits each vertex (representing a player) of V exactly once.

Theorem: Every tournament has a Hamiltonian path.

Proof will be by the induction on the number n of vertices in a tournament.

Induction step. Suppose that every tournament with at most n vertices has a Hamiltonian path and let a tournament $T = (V, E)$ with $n + 1$ vertices be given.

Choose any vertex v and define two sets of vertices

$$A = \{u \mid (u, v) \in E\} \qquad B = \{u \mid (v, u) \in E\}.$$

Two subgraphs induced by these two sets of vertices form tournaments. By induction both of them have Hamiltonian paths.

HAMILTONIAN PATHS in TOURNAMENTS

A **tournament** is a complete directed graph.

A **Hamiltonian path** in a graph $G = (V, E)$ is a path that visits each vertex (representing a player) of V exactly once.

Theorem: Every tournament has a Hamiltonian path.

Proof will be by the induction on the number n of vertices in a tournament.

Induction step. Suppose that every tournament with at most n vertices has a Hamiltonian path and let a tournament $T = (V, E)$ with $n + 1$ vertices be given.

Choose any vertex v and define two sets of vertices

$$A = \{u \mid (u, v) \in E\} \qquad B = \{u \mid (v, u) \in E\}.$$

Two subgraphs induced by these two sets of vertices form tournaments. By induction both of them have Hamiltonian paths.

By connecting these two paths through the node v we get a Hamiltonian path for the tournament T.

NUMBER of PATHS in TOURNAMENTS

NUMBER of PATHS in TOURNAMENTS

Theorem: For any n there exists a tournament of size n with $n!/2^{n-1}$

NUMBER of PATHS in TOURNAMENTS

Theorem: For any n there exists a tournament of size n with $n!/2^{n-1}$ Hamiltonian paths.

NUMBER of PATHS in TOURNAMENTS

Theorem: For any n there exists a tournament of size n with $n!/2^{n-1}$ Hamiltonian paths.

Proof: Generate a random tournament $T = (V, E)$, $V = \{1, \ldots, n\}$, by randomly choosing direction for all edges of K_n - of a complete graph of n vertices.

NUMBER of PATHS in TOURNAMENTS

Theorem: For any n there exists a tournament of size n with $n!/2^{n-1}$ Hamiltonian paths.

Proof: Generate a random tournament $T = (V, E)$, $V = \{1, \ldots, n\}$, by randomly choosing direction for all edges of K_n - of a complete graph of n vertices.

For each permutation σ on V let X_σ be a random variable defined as follows:

NUMBER of PATHS in TOURNAMENTS

Theorem: For any n there exists a tournament of size n with $n!/2^{n-1}$ Hamiltonian paths.

Proof: Generate a random tournament $T = (V, E)$, $V = \{1, \ldots, n\}$, by randomly choosing direction for all edges of K_n - of a complete graph of n vertices.

For each permutation σ on V let X_σ be a random variable defined as follows:

$$X_\sigma = \begin{cases} 1, & \sigma \text{ defines a path on } T \\ 0 & \text{otherwise} \end{cases}$$

NUMBER of PATHS in TOURNAMENTS

Theorem: For any n there exists a tournament of size n with $n!/2^{n-1}$ Hamiltonian paths.

Proof: Generate a random tournament $T = (V, E)$, $V = \{1, \ldots, n\}$, by randomly choosing direction for all edges of K_n - of a complete graph of n vertices.

For each permutation σ on V let X_σ be a random variable defined as follows:

$$X_\sigma = \begin{cases} 1, & \sigma \text{ defines a path on } T \\ 0 & \text{otherwise} \end{cases}$$

For all σ, $\Pr(X_\sigma = 1) = (\frac{1}{2})^{n-1} = \mathbf{E}[X_\sigma]$.

NUMBER of PATHS in TOURNAMENTS

Theorem: For any n there exists a tournament of size n with $n!/2^{n-1}$ Hamiltonian paths.

Proof: Generate a random tournament $T = (V, E)$, $V = \{1, \ldots, n\}$, by randomly choosing direction for all edges of K_n - of a complete graph of n vertices.

For each permutation σ on V let X_σ be a random variable defined as follows:

$$X_\sigma = \begin{cases} 1, & \sigma \text{ defines a path on } T \\ 0 & \text{otherwise} \end{cases}$$

For all σ, $\Pr(X_\sigma = 1) = (\frac{1}{2})^{n-1} = \mathbf{E}[X_\sigma]$.

Let X be a random variable counting the number of Hamiltonian paths in T

NUMBER of PATHS in TOURNAMENTS

Theorem: For any n there exists a tournament of size n with $n!/2^{n-1}$ Hamiltonian paths.

Proof: Generate a random tournament $T = (V, E)$, $V = \{1, \ldots, n\}$, by randomly choosing direction for all edges of K_n - of a complete graph of n vertices.

For each permutation σ on V let X_σ be a random variable defined as follows:

$$X_\sigma = \begin{cases} 1, & \sigma \text{ defines a path on } T \\ 0 & \text{otherwise} \end{cases}$$

For all σ, $\Pr(X_\sigma = 1) = (\frac{1}{2})^{n-1} = \mathbf{E}[X_\sigma]$.

Let X be a random variable counting the number of Hamiltonian paths in T

$$X = \sum_{\sigma \in \mathrm{Perm}(n)} X_\sigma .$$

NUMBER of PATHS in TOURNAMENTS

Theorem: For any n there exists a tournament of size n with $n!/2^{n-1}$ Hamiltonian paths.

Proof: Generate a random tournament $T = (V, E)$, $V = \{1, \ldots, n\}$, by randomly choosing direction for all edges of K_n - of a complete graph of n vertices.

For each permutation σ on V let X_σ be a random variable defined as follows:

$$X_\sigma = \begin{cases} 1, & \sigma \text{ defines a path on } T \\ 0 & \text{otherwise} \end{cases}$$

For all σ, $\Pr(X_\sigma = 1) = (\frac{1}{2})^{n-1} = \mathbf{E}[X_\sigma]$.

Let X be a random variable counting the number of Hamiltonian paths in T

$$X = \sum_{\sigma \in \mathrm{Perm}(n)} X_\sigma.$$

Theorem now follows from the following calculations:

$$\mathbf{E}[X] = \mathbf{E}[\sum_\sigma X_\sigma] = \sum_\sigma \mathbf{E}[X_\sigma] = n!(\frac{1}{2})^{n-1}.$$

TOURNAMENTS WITH a PROPERTY S_k

TOURNAMENTS WITH a PROPERTY S_k

In a tournament, if there is an edge from a node A to a node B, then we say that the player A beats the player B.

TOURNAMENTS WITH a PROPERTY S_k

In a tournament, if there is an edge from a node A to a node B, then we say that the player A beats the player B.

A tournament T is said to have property S_k, for an integer k, if for any set of k players there is one player that beats all of them.

TOURNAMENTS WITH a PROPERTY S_k

In a tournament, if there is an edge from a node A to a node B, then we say that the player A beats the player B.

A tournament T is said to have property S_k, for an integer k, if for any set of k players there is one player that beats all of them.

Theorem If $\binom{n}{k}(1-2^{-k})^{n-k} < 1$, then there is a tournament on n vertices with property S_k.

TOURNAMENTS WITH a PROPERTY S_k

In a tournament, if there is an edge from a node A to a node B, then we say that the player A beats the player B.

A tournament T is said to have property S_k, for an integer k, if for any set of k players there is one player that beats all of them.

Theorem If $\binom{n}{k}(1-2^{-k})^{n-k} < 1$, then there is a tournament on n vertices with property S_k. **Proof** Consider a random tournament with a set of n nodes.

TOURNAMENTS WITH a PROPERTY S_k

In a tournament, if there is an edge from a node A to a node B, then we say that the player A beats the player B.

A tournament T is said to have property S_k, for an integer k, if for any set of k players there is one player that beats all of them.

Theorem If $\binom{n}{k}(1-2^{-k})^{n-k} < 1$, then there is a tournament on n vertices with property S_k. **Proof** Consider a random tournament with a set of n nodes. For any subset K of k vertices let A_K be the event that there is no player/node that beats all players/nodes in K. Clearly,

TOURNAMENTS WITH a PROPERTY S_k

In a tournament, if there is an edge from a node A to a node B, then we say that the player A beats the player B.
A tournament T is said to have property S_k, for an integer k, if for any set of k players there is one player that beats all of them.

Theorem If $\binom{n}{k}(1-2^{-k})^{n-k} < 1$, then there is a tournament on n vertices with property S_k. **Proof** Consider a random tournament with a set of n nodes. For any subset K of k vertices let A_K be the event that there is no player/node that beats all players/nodes in K. Clearly,

$$\Pr[A_K] = (1-2^{-k})^{n-k}.$$

TOURNAMENTS WITH a PROPERTY S_k

In a tournament, if there is an edge from a node A to a node B, then we say that the player A beats the player B.
A tournament T is said to have property S_k, for an integer k, if for any set of k players there is one player that beats all of them.

Theorem If $\binom{n}{k}(1-2^{-k})^{n-k} < 1$, then there is a tournament on n vertices with property S_k. **Proof** Consider a random tournament with a set of n nodes. For any subset K of k vertices let A_K be the event that there is no player/node that beats all players/nodes in K. Clearly,

$$\Pr[A_K] = (1-2^{-k})^{n-k}.$$

Therefore

$$\Pr[\bigvee_{K\subset V, |K|=k} A_K] \leq \sum_{K\subset V, |K|=k} \Pr[A_K] = \binom{n}{k}(1-2^{-k})^{n-k} < 1$$

TOURNAMENTS WITH a PROPERTY S_k

In a tournament, if there is an edge from a node A to a node B, then we say that the player A beats the player B.
A tournament T is said to have property S_k, for an integer k, if for any set of k players there is one player that beats all of them.

Theorem If $\binom{n}{k}(1-2^{-k})^{n-k} < 1$, then there is a tournament on n vertices with property S_k. **Proof** Consider a random tournament with a set of n nodes. For any subset K of k vertices let A_K be the event that there is no player/node that beats all players/nodes in K. Clearly,

$$\Pr[A_K] = (1-2^{-k})^{n-k}.$$

Therefore

$$\Pr[\bigvee_{K \subset V, |K|=k} A_K] \leq \sum_{K \subset V, |K|=k} \Pr[A_K] = \binom{n}{k}(1-2^{-k})^{n-k} < 1$$

Therefore, with a positive probability no event A_K occurs. That is, there is a tournament on n vertices that has the property S_k.

EXPLANATION

EXPLANATION

If K is a set on k players, then the probability that a player P not in K beats all of them is 2^{-k}

EXPLANATION

If K is a set on k players, then the probability that a player P not in K beats all of them is 2^{-k} and the probability is $1 - 2^{-k}$ that he does not beat all of them.

EXPLANATION

If K is a set on k players, then the probability that a player P not in K beats all of them is 2^{-k} and the probability is $1 - 2^{-k}$ that he does not beat all of them.

Since there are $n - k$ players outside the group K, the probability that none of them beats all players in K is $(1 - 2^{-k})^{n-k}$.

MAX-SAT - PROBLEM

MAX-SAT - PROBLEM

Given are m clauses in conjunctive normal form over n variables.

MAX-SAT - PROBLEM

Given are m clauses in conjunctive normal form over n variables. Find assignment (of truth values to variables) that maximizes the number of satisfied clauses.

MAX-SAT - PROBLEM

Given are m clauses in conjunctive normal form over n variables. Find assignment (of truth values to variables) that maximizes the number of satisfied clauses.

Theorem: For any set of m clauses there is a truth assignment that satisfies at least $\frac{m}{2}$ clauses.

MAX-SAT - PROBLEM

Given are m clauses in conjunctive normal form over n variables. Find assignment (of truth values to variables) that maximizes the number of satisfied clauses.

Theorem: For any set of m clauses there is a truth assignment that satisfies at least $\frac{m}{2}$ clauses.

Proof: Suppose each variable is set to 0 or 1 independently and equiprobably and let, for $1 \leq i \leq m$, the random variable $Z_i = 1$ if the i-th clause is satisfied.

MAX-SAT - PROBLEM

Given are m clauses in conjunctive normal form over n variables. Find assignment (of truth values to variables) that maximizes the number of satisfied clauses.

Theorem: For any set of m clauses there is a truth assignment that satisfies at least $\frac{m}{2}$ clauses.

Proof: Suppose each variable is set to 0 or 1 independently and equiprobably and let, for $1 \leq i \leq m$, the random variable $Z_i = 1$ if the i-th clause is satisfied.

The probability that a clause with k literals is not satisfied by this random assignment is 2^{-k}.

MAX-SAT - PROBLEM

Given are m clauses in conjunctive normal form over n variables. Find assignment (of truth values to variables) that maximizes the number of satisfied clauses.

Theorem: For any set of m clauses there is a truth assignment that satisfies at least $\frac{m}{2}$ clauses.

Proof: Suppose each variable is set to 0 or 1 independently and equiprobably and let, for $1 \leq i \leq m$, the random variable $Z_i = 1$ if the i-th clause is satisfied.

The probability that a clause with k literals is not satisfied by this random assignment is 2^{-k}.

The probability that a clause with k literals is satisfied is $1 - \frac{1}{2^k} \geq \frac{1}{2}$ what implies that $\mathbf{E}[Z_i] \geq \frac{1}{2}$ for all i.

MAX-SAT - PROBLEM

Given are m clauses in conjunctive normal form over n variables. Find assignment (of truth values to variables) that maximizes the number of satisfied clauses.

Theorem: For any set of m clauses there is a truth assignment that satisfies at least $\frac{m}{2}$ clauses.

Proof: Suppose each variable is set to 0 or 1 independently and equiprobably and let, for $1 \leq i \leq m$, the random variable $Z_i = 1$ if the i-th clause is satisfied.

The probability that a clause with k literals is not satisfied by this random assignment is 2^{-k}.

The probability that a clause with k literals is satisfied is $1 - \frac{1}{2^k} \geq \frac{1}{2}$ what implies that $\mathbf{E}[Z_i] \geq \frac{1}{2}$ for all i.

The expected number of clauses satisfied by a random assignment is

$$\mathbf{E}[\sum_{i=1}^{m} Z_i] = \sum_{i=1}^{m} \mathbf{E}[Z_i] \geq \frac{m}{2}$$

MAX-SAT - PROBLEM

Given are m clauses in conjunctive normal form over n variables. Find assignment (of truth values to variables) that maximizes the number of satisfied clauses.

Theorem: For any set of m clauses there is a truth assignment that satisfies at least $\frac{m}{2}$ clauses.

Proof: Suppose each variable is set to 0 or 1 independently and equiprobably and let, for $1 \leq i \leq m$, the random variable $Z_i = 1$ if the i-th clause is satisfied.

The probability that a clause with k literals is not satisfied by this random assignment is 2^{-k}.

The probability that a clause with k literals is satisfied is $1 - \frac{1}{2^k} \geq \frac{1}{2}$ what implies that $\mathbf{E}[Z_i] \geq \frac{1}{2}$ for all i.

The expected number of clauses satisfied by a random assignment is

$$\mathbf{E}[\sum_{i=1}^{m} Z_i] = \sum_{i=1}^{m} \mathbf{E}[Z_i] \geq \frac{m}{2}$$

Therefore, there exists at least one assignment for which $\sum_{i=1}^{m} Z_i \geq \frac{m}{2}$.

FROM THE PROOF of EXISTENCE to an ALGORITHM - I.

FROM THE PROOF of EXISTENCE to an ALGORITHM - I.

We show now that a variant of the probabilistic proof of existence in the last theorem can be turned into an approximation algorithm.

FROM THE PROOF of EXISTENCE to an ALGORITHM - I.

We show now that a variant of the probabilistic proof of existence in the last theorem can be turned into an approximation algorithm.

Notation for approximation algorithms for MAX-SAT problem:

FROM THE PROOF of EXISTENCE to an ALGORITHM - I.

We show now that a variant of the probabilistic proof of existence in the last theorem can be turned into an approximation algorithm.

Notation for approximation algorithms for MAX-SAT problem:

- I - a particular input instance - a set of clauses.

FROM THE PROOF of EXISTENCE to an ALGORITHM - I.

We show now that a variant of the probabilistic proof of existence in the last theorem can be turned into an approximation algorithm.

Notation for approximation algorithms for MAX-SAT problem:

- I - a particular input instance - a set of clauses.
- $m_*(I)$ - the maximum number of clauses of I that can be satisfied.

FROM THE PROOF of EXISTENCE to an ALGORITHM - I.

We show now that a variant of the probabilistic proof of existence in the last theorem can be turned into an approximation algorithm.

Notation for approximation algorithms for MAX-SAT problem:

- I - a particular input instance - a set of clauses.
- $m_*(I)$ - the maximum number of clauses of I that can be satisfied.
- $m_A(I)$ - the number of clauses of I satisfied by an algorithm A.

FROM THE PROOF of EXISTENCE to an ALGORITHM - I.

We show now that a variant of the probabilistic proof of existence in the last theorem can be turned into an approximation algorithm.

Notation for approximation algorithms for MAX-SAT problem:

- I - a particular input instance - a set of clauses.
- $m_*(I)$ - the maximum number of clauses of I that can be satisfied.
- $m_A(I)$ - the number of clauses of I satisfied by an algorithm A.
- **performance ratio** of an algorithm A : $\inf_I \frac{m_A(I)}{m_*(I)}$.

FROM THE PROOF of EXISTENCE to an ALGORITHM - I.

We show now that a variant of the probabilistic proof of existence in the last theorem can be turned into an approximation algorithm.

Notation for approximation algorithms for MAX-SAT problem:

- I - a particular input instance - a set of clauses.
- $m_*(I)$ - the maximum number of clauses of I that can be satisfied.
- $m_A(I)$ - the number of clauses of I satisfied by an algorithm A.
- **performance ratio** of an algorithm A : $\inf_I \frac{m_A(I)}{m_*(I)}$.

If A achieves a performance ratio α, we say that A is an α-approximation algorithm.

FROM THE PROOF of EXISTENCE to an ALGORITHM - I.

We show now that a variant of the probabilistic proof of existence in the last theorem can be turned into an approximation algorithm.

Notation for approximation algorithms for MAX-SAT problem:

- I - a particular input instance - a set of clauses.
- $m_*(I)$ - the maximum number of clauses of I that can be satisfied.
- $m_A(I)$ - the number of clauses of I satisfied by an algorithm A.
- **performance ratio** of an algorithm A: $\inf_I \frac{m_A(I)}{m_*(I)}$.

If A achieves a performance ratio α, we say that A is an α-approximation algorithm.

If A is a randomized algorithm, then $m_A(I)$ is a random variable and in such a case $m_A(I)$ is replaced by $\mathbf{E}[m_A(I)]$ in the definition of the performance ratio.

FROM a PROOF of EXISTENCE to an ALGORITHM - II.

FROM a PROOF of EXISTENCE to an ALGORITHM - II.

We now show the existence of a randomized algorithm for MAXSAT with performance ratio $\frac{3}{4}$.

FROM a PROOF of EXISTENCE to an ALGORITHM - II.

We now show the existence of a randomized algorithm for MAXSAT with performance ratio $\frac{3}{4}$.

The procedure in the proof of the last theorem actually yields a randomized algorithm whose guaranteed performance is $1 - 2^{-k}$, provided every clause contains at least k literals.

FROM a PROOF of EXISTENCE to an ALGORITHM - II.

We now show the existence of a randomized algorithm for MAXSAT with performance ratio $\frac{3}{4}$.

The procedure in the proof of the last theorem actually yields a randomized algorithm whose guaranteed performance is $1 - 2^{-k}$, provided every clause contains at least k literals.

As a consequence, *we have a randomized $\frac{3}{4}$-approximation algorithm for instances of MAX-SAT in which every clause contains at least 2 literals.*

FROM a PROOF of EXISTENCE to an ALGORITHM - II.

We now show the existence of a randomized algorithm for MAXSAT with performance ratio $\frac{3}{4}$.

The procedure in the proof of the last theorem actually yields a randomized algorithm whose guaranteed performance is $1 - 2^{-k}$, provided every clause contains at least k literals.

As a consequence, *we have a randomized $\frac{3}{4}$-approximation algorithm for instances of MAX-SAT in which every clause contains at least 2 literals.*

We now show another algorithm that performs especially well when there are (many) clauses consisting of a single literal.

FROM a PROOF of EXISTENCE to an ALGORITHM - II.

We now show the existence of a randomized algorithm for MAXSAT with performance ratio $\frac{3}{4}$.

The procedure in the proof of the last theorem actually yields a randomized algorithm whose guaranteed performance is $1 - 2^{-k}$, provided every clause contains at least k literals.

As a consequence, *we have a randomized $\frac{3}{4}$-approximation algorithm for instances of MAX-SAT in which every clause contains at least 2 literals.*

We now show another algorithm that performs especially well when there are (many) clauses consisting of a single literal.

Finally, we show that on any input instance, one of the two designed algorithms yields a randomized $\frac{3}{4}$-approximation algorithm.

BASIC IDEA

BASIC IDEA

is similar as in the case of the global wiring problem.

BASIC IDEA

is similar as in the case of the global wiring problem.

1. Reformulate the problem as a 0-1 linear programming problem.

BASIC IDEA

is similar as in the case of the global wiring problem.

1. Reformulate the problem as a 0-1 linear programming problem.
2. Solve the corresponding rational linear programming problem.

BASIC IDEA

is similar as in the case of the global wiring problem.

1. Reformulate the problem as a 0-1 linear programming problem.
2. Solve the corresponding rational linear programming problem.
3. Use the randomized rounding technique.

BASIC IDEA

is similar as in the case of the global wiring problem.

1. Reformulate the problem as a 0-1 linear programming problem.
2. Solve the corresponding rational linear programming problem.
3. Use the randomized rounding technique.

Notation: *With each clause C_j, in the given input formula, we associate an indicator variable $c_j \in \{0, 1\}$, that indicates whether or not the clause C_j is satisfied at the algorithm being used.*

INDICATOR VARIABLES

INDICATOR VARIABLES

Moreover, to each variable x_i we assign an indicator variable v_i defined by

$$x_i = true \iff v_i = 1 \qquad .$$

C_j^+ - set of indices of variables that appear uncomplemented in C_j
C_j^- - set of indices of variables that appear complemented in C_j

0-1 linear programming reformulation of the MAX-SAT problem.

0-1 linear programming reformulation of the MAX-SAT problem.

Find

$$v_i, c_j \in \{0, 1\} \quad (\forall i, j) \qquad\qquad (\star)$$

0-1 linear programming reformulation of the MAX-SAT problem.

Find

$$v_i, c_j \in \{0, 1\} \quad (\forall i, j) \qquad (\star)$$

such that the sum

$$\sum_{j=1}^{m} c_j$$

is maximized

0-1 linear programming reformulation of the MAX-SAT problem.

Find

$$v_i, c_j \in \{0,1\} \quad (\forall i, j) \qquad\qquad (\star)$$

such that the sum

$$\sum_{j=1}^{m} c_j$$

is maximized and

$$\sum_{i \in C_j^+} v_i + \sum_{i \in C_j^-} (1 - v_i) \geq c_j \quad (\forall j). \tag{1}$$

0-1 linear programming reformulation of the MAX-SAT problem.

Find

$$v_i, c_j \in \{0,1\} \quad (\forall i, j) \qquad\qquad (\star)$$

such that the sum

$$\sum_{j=1}^{m} c_j$$

is maximized and

$$\sum_{i \in C_j^+} v_i + \sum_{i \in C_j^-} (1 - v_i) \geq c_j \quad (\forall j). \qquad (1)$$

Rational linear programming problem is then obtained by replacing the condition $(\star)$ by the condition

$$v_i, c_j \in [0,1] \quad (\forall i, j).$$

0-1 linear programming reformulation of the MAX-SAT problem.

Find

$$v_i, c_j \in \{0,1\} \quad (\forall i, j) \qquad (\star)$$

such that the sum

$$\sum_{j=1}^{m} c_j$$

is maximized and

$$\sum_{i \in C_j^+} v_i + \sum_{i \in C_j^-} (1 - v_i) \geq c_j \quad (\forall j). \tag{1}$$

Rational linear programming problem is then obtained by replacing the condition $(\star)$ by the condition

$$v_i, c_j \in [0,1] \quad (\forall i, j).$$

Let $\hat{v}_i$ ($\hat{c}_i$) be the value of variable v_i (c_i) obtained by solving the rational linear programming problem. Clearly, $\sum_{i=1}^{n} c_j \leq \sum_{j=1}^{n} \hat{c}_j$.

CONTINUATION of THE PROOF 1/2

CONTINUATION of THE PROOF 1/2

We first show that using the randomized rounding method we obtain a truth assignment for which the expected number of satisfied clauses is at least $(1 - \frac{1}{e}) \sum_j \hat{c}_j$.

CONTINUATION of THE PROOF 1/2

We first show that using the randomized rounding method we obtain a truth assignment for which the expected number of satisfied clauses is at least $(1 - \frac{1}{e}) \sum_j \hat{c}_j$.

This will follow from the Lemma shown on the next slide for the case we use the following **randomized rounding:** each v_i is set, independently, to 1 with the probability $\hat{v}_i$.

CONTINUATION of THE PROOF 1/2

We first show that using the randomized rounding method we obtain a truth assignment for which the expected number of satisfied clauses is at least $(1 - \frac{1}{e})\sum_{j} \hat{c}_j$.

This will follow from the Lemma shown on the next slide for the case we use the following **randomized rounding:** each v_i is set, independently, to 1 with the probability $\hat{v}_i$.

Notation: For an integer k denote $\beta_k = 1 - (1 - \frac{1}{k})^k > 1 - \frac{1}{e}$.

CONTINUATION of THE PROOF 2/2

CONTINUATION of THE PROOF 2/2

Lemma: Let C_j be a clause with k literals. The probability that it is satisfied by the randomized rounding is at least $\beta_k \hat{c}_j > (1 - \frac{1}{e})\hat{c}_j$.

CONTINUATION of THE PROOF 2/2

Lemma: Let C_j be a clause with k literals. The probability that it is satisfied by the randomized rounding is at least $\beta_k \hat{c}_j > (1 - \frac{1}{e})\hat{c}_j$.

Proof: Without loss of generality we can assume that

$$C_j : \quad x_1 \vee \cdots \vee x_k$$

CONTINUATION of THE PROOF 2/2

Lemma: Let C_j be a clause with k literals. The probability that it is satisfied by the randomized rounding is at least $\beta_k \hat{c}_j > (1 - \frac{1}{e})\hat{c}_j$.

Proof: Without loss of generality we can assume that

$$C_j : \quad x_1 \vee \cdots \vee x_k$$

By constrain (1) $\qquad \hat{v}_1 + \cdots + \hat{v}_k \geq \hat{c}_j.$

CONTINUATION of THE PROOF 2/2

Lemma: Let C_j be a clause with k literals. The probability that it is satisfied by the randomized rounding is at least $\beta_k \hat{c}_j > (1 - \frac{1}{e})\hat{c}_j$.

Proof: Without loss of generality we can assume that

$$C_j : \quad x_1 \vee \cdots \vee x_k$$

By constrain (1) $\qquad \hat{v}_1 + \cdots + \hat{v}_k \geq \hat{c}_j$.
Observe that the clause C_j remains unsatisfied by randomized rounding method only if every variable v_i is rounded to 0.

CONTINUATION of THE PROOF 2/2

Lemma: Let C_j be a clause with k literals. The probability that it is satisfied by the randomized rounding is at least $\beta_k \hat{c}_j > (1 - \frac{1}{e})\hat{c}_j$.

Proof: Without loss of generality we can assume that

$$C_j : \quad x_1 \vee \cdots \vee x_k$$

By constrain (1) $\quad \hat{v}_1 + \cdots + \hat{v}_k \geq \hat{c}_j$.
Observe that the clause C_j remains unsatisfied by randomized rounding method only if every variable v_i is rounded to 0.
Since each variable is rounded fully independently, this occurs with probability

$$\prod_{i=1}^{k} (1 - \hat{v}_i).$$

CONTINUATION of THE PROOF 2/2

Lemma: Let C_j be a clause with k literals. The probability that it is satisfied by the randomized rounding is at least $\beta_k \hat{c}_j > (1 - \frac{1}{e})\hat{c}_j$.

Proof: Without loss of generality we can assume that

$$C_j : \quad x_1 \vee \cdots \vee x_k$$

By constrain (1) $\qquad \hat{v}_1 + \cdots + \hat{v}_k \geq \hat{c}_j$.
Observe that the clause C_j remains unsatisfied by randomized rounding method only if every variable v_i is rounded to 0.
Since each variable is rounded fully independently, this occurs with probability

$$\prod_{i=1}^{k}(1 - \hat{v}_i).$$

It remains to show that

$$\underbrace{1 - \prod_{i=1}^{k}(1 - \hat{v}_i)}$$

CONTINUATION of THE PROOF 2/2

Lemma: Let C_j be a clause with k literals. The probability that it is satisfied by the randomized rounding is at least $\beta_k \hat{c}_j > (1 - \frac{1}{e})\hat{c}_j$.

Proof: Without loss of generality we can assume that

$$C_j : \quad x_1 \vee \cdots \vee x_k$$

By constrain (1) $\qquad \hat{v}_1 + \cdots + \hat{v}_k \geq \hat{c}_j$.

Observe that the clause C_j remains unsatisfied by randomized rounding method only if every variable v_i is rounded to 0.

Since each variable is rounded fully independently, this occurs with probability

$$\prod_{i=1}^{k} (1 - \hat{v}_i).$$

It remains to show that

$$\underbrace{1 - \prod_{i=1}^{k} (1 - \hat{v}_i)}_{\textit{left side is minimized if } \hat{v}_i = \frac{\hat{c}_j}{k}} \qquad \geq \; \beta_k \hat{c}_j$$

Again: It remains to show that

$$\underbrace{1 - \prod_{i=1}^{k} (1 - \hat{v}_i)}$$

Again: It remains to show that

$$\underbrace{1 - \prod_{i=1}^{k}(1 - \hat{v}_i)}_{andtheleftsideisminimizedif\, \hat{v}_i = \frac{\hat{c}_j}{k}} \qquad \geq \ \beta_k \hat{c}_j$$

This can be shown if one can show that $1 - (1 - \frac{c}{k})^k \geq \beta_k c$ *for all* $0 < z < 1$.

Again: It remains to show that

$$\underbrace{1-\prod_{i=1}^{k}(1-\hat{v}_i)}_{andtheleftsideisminimizedif\,\hat{v}_i=\frac{\hat{c}_j}{k}} \quad \geq \ \beta_k \hat{c}_j$$

This can be shown if one can show that $1-(1-\frac{c}{k})^k \geq \beta_k c$ *for all* $0 < z < 1$.

Since function $f(x) = 1-(1-\frac{x}{k})^k$ is concave, it suffices to verify the above inequality for $x = 0$ and $x = 1$ what is easy.

FINAL RESULT I

FINAL RESULT I

From the last Lemma, and from the linearity of expectations, it follows:

FINAL RESULT I

From the last Lemma, and from the linearity of expectations, it follows:

Theorem: Given an instance of MAX-SAT, the expected number of clauses satisfied by linear programming and randomized rounding is at least $(1 - \frac{1}{e})$ time the maximum number of clauses that can be satisfied on that instance.

FINAL RESULT I

From the last Lemma, and from the linearity of expectations, it follows:

Theorem: Given an instance of MAX-SAT, the expected number of clauses satisfied by linear programming and randomized rounding is at least $(1 - \frac{1}{e})$ time the maximum number of clauses that can be satisfied on that instance.

Comparison of performances of our two algorithms for MAX-SAT

FINAL RESULT I

From the last Lemma, and from the linearity of expectations, it follows:

Theorem: Given an instance of MAX-SAT, the expected number of clauses satisfied by linear programming and randomized rounding is at least $(1 - \frac{1}{e})$ time the maximum number of clauses that can be satisfied on that instance.

Comparison of performances of our two algorithms for MAX-SAT

k	$1 - 2^{-k}$	β_k
1	0.5	1.0
2	0.75	0.75
3	0.875	0.704
4	0.938	0.684
5	0.969	0.672

FINAL RESULT I

From the last Lemma, and from the linearity of expectations, it follows:

Theorem: Given an instance of MAX-SAT, the expected number of clauses satisfied by linear programming and randomized rounding is at least $(1 - \frac{1}{e})$ time the maximum number of clauses that can be satisfied on that instance.

Comparison of performances of our two algorithms for MAX-SAT

k	$1 - 2^{-k}$	β_k
1	0.5	1.0
2	0.75	0.75
3	0.875	0.704
4	0.938	0.684
5	0.969	0.672

We now show that on any instance one of the algorithms is a $\frac{3}{4}$ - approximation algorithm for the MAX-SAT problem:

Let n_1 denote the expected number of clauses that are satisfied when each variable is independently set to 1 with probability $\frac{1}{2}$ (what corresponds to the first algorithm).

Let n_1 denote the expected number of clauses that are satisfied when each variable is independently set to 1 with probability $\frac{1}{2}$ (what corresponds to the first algorithm).

Let n_2 denote the expected number of clauses that are satisfied when we use the linear programming followed by the randomized rounding (what corresponds to the second algorithm).

FINAL RESULT II

FINAL RESULT II

Theorem:

$$\max\{n_1, n_2\} \geq \frac{3}{4}\sum_j \hat{c}_j.$$

FINAL RESULT II

Theorem:

$$\max\{n_1, n_2\} \geq \frac{3}{4} \sum_j \hat{c}_j.$$

Proof: It suffices to show that $\left(\frac{n_1+n_2}{2}\right) \geq \frac{3}{4} \sum_j \hat{c}_j$.

FINAL RESULT II

Theorem:

$$\max\{n_1, n_2\} \geq \frac{3}{4} \sum_j \hat{c}_j.$$

Proof: It suffices to show that $\left(\frac{n_1+n_2}{2}\right) \geq \frac{3}{4} \sum_j \hat{c}_j$.

Let S_k denote the set of clauses that contain k literals. We know that

$$n_1 = \sum_k \sum_{C_j \in S_k} (1 - 2^{-k}) c_j \geq \sum_k \sum_{C_j \in S_k} (1 - 2^{-k}) \hat{c}_j,$$

$$n_2 \geq \sum_k \sum_{C_j \in S_k} \beta_k \hat{c}_j.$$

FINAL RESULT II

Theorem:

$$\max\{n_1, n_2\} \geq \frac{3}{4} \sum_j \hat{c}_j.$$

Proof: It suffices to show that $(\frac{n_1+n_2}{2}) \geq \frac{3}{4} \sum_j \hat{c}_j$.

Let S_k denote the set of clauses that contain k literals. We know that

$$n_1 = \sum_k \sum_{C_j \in S_k} (1 - 2^{-k}) c_j \geq \sum_k \sum_{C_j \in S_k} (1 - 2^{-k}) \hat{c}_j,$$

$$n_2 \geq \sum_k \sum_{C_j \in S_k} \beta_k \hat{c}_j.$$

Thus

$$\frac{n_1 + n_2}{2} \geq \sum_k \sum_{C_j \in S_k} \frac{(1 - 2^{-k}) + \beta_k}{2} \hat{c}_j.$$

FINAL RESULT II

Theorem:

$$\max\{n_1, n_2\} \geq \frac{3}{4} \sum_j \hat{c}_j.$$

Proof: It suffices to show that $\left(\frac{n_1+n_2}{2}\right) \geq \frac{3}{4} \sum_j \hat{c}_j$.

Let S_k denote the set of clauses that contain k literals. We know that

$$n_1 = \sum_k \sum_{C_j \in S_k} (1 - 2^{-k}) c_j \geq \sum_k \sum_{C_j \in S_k} (1 - 2^{-k}) \hat{c}_j,$$

$$n_2 \geq \sum_k \sum_{C_j \in S_k} \beta_k \hat{c}_j.$$

Thus

$$\frac{n_1 + n_2}{2} \geq \sum_k \sum_{C_j \in S_k} \frac{(1 - 2^{-k}) + \beta_k}{2} \hat{c}_j.$$

Since $(1 - 2^{-k}) + \beta_k \geq \frac{3}{2}$ for all k, we get

$$\frac{n_1 + n_2}{2} \geq \frac{3}{4} \sum_k \sum_{C_j \in S_k} \hat{c}_j = \frac{3}{4} \sum_j \hat{c}_j.$$

RAMSEY NUMBER PROBLEM

RAMSEY NUMBER PROBLEM

The **Ramsey number** $R(k, l)$ is the smallest integer n such that in any 2-coloring of the edges of a complete graph K_n, on n nodes, by red and blue, there either is a red K_k (i.e. a complete subgraph on k vertices with edges coloured red), or there is a blue K_l.

RAMSEY NUMBER PROBLEM

The **Ramsey number** $R(k, l)$ is the smallest integer n such that in any 2-coloring of the edges of a complete graph K_n, on n nodes, by red and blue, there either is a red K_k (i.e. a complete subgraph on k vertices with edges coloured red), or there is a blue K_l.

Ramsey (1930) showed that $R(k, l)$ is finite for any two integers k and l.

RAMSEY NUMBER PROBLEM

The **Ramsey number** $R(k, l)$ is the smallest integer n such that in any 2-coloring of the edges of a complete graph K_n, on n nodes, by red and blue, there either is a red K_k (i.e. a complete subgraph on k vertices with edges coloured red), or there is a blue K_l.

Ramsey (1930) showed that $R(k, l)$ is finite for any two integers k and l.

We use the probabilistic method to show a lower bound on $R(k, k)$.

A LOWER BOUND on RAMSEY NUMBERS

A LOWER BOUND on RAMSEY NUMBERS

Theorem: If $\binom{n}{k} \cdot 2^{1-\binom{k}{2}} < 1$, then $R(k,k) > n$.

A LOWER BOUND on RAMSEY NUMBERS

Theorem: If $\binom{n}{k} \cdot 2^{1-\binom{k}{2}} < 1$, then $R(k,k) > n$.

Corollary Since $\binom{n}{k} \cdot 2^{1-\binom{k}{2}} < 1$ for $n = 2^{k/2}$ (that is if $k = 2\lg n$), the above theorem implies that $R(k,k) > 2^{k/2}$ for all $k \geq 3$.

A LOWER BOUND on RAMSEY NUMBERS

Theorem: If $\binom{n}{k} \cdot 2^{1-\binom{k}{2}} < 1$, then $R(k,k) > n$.

Corollary Since $\binom{n}{k} \cdot 2^{1-\binom{k}{2}} < 1$ for $n = 2^{k/2}$ (that is if $k = 2\lg n$), the above theorem implies that $R(k,k) > 2^{k/2}$ for all $k \geq 3$.

Proof of Theorem: Let n, k satisfy the assumption of the theorem. Consider a random 2-coloring of the edges of K_n by red or blue.

A LOWER BOUND on RAMSEY NUMBERS

Theorem: If $\binom{n}{k} \cdot 2^{1-\binom{k}{2}} < 1$, then $R(k,k) > n$.

Corollary Since $\binom{n}{k} \cdot 2^{1-\binom{k}{2}} < 1$ for $n = 2^{k/2}$ (that is if $k = 2\lg n$), the above theorem implies that $R(k,k) > 2^{k/2}$ for all $k \geq 3$.

Proof of Theorem: Let n, k satisfy the assumption of the theorem. Consider a random 2-coloring of the edges of K_n by red or blue.

For any fixed set R of k vertices, let A_R be the event that the induced subgraph of K_n on R is **monochromatic** (i.e. that either all its edges are red or they are all blue).

A LOWER BOUND on RAMSEY NUMBERS

Theorem: If $\binom{n}{k} \cdot 2^{1-\binom{k}{2}} < 1$, then $R(k,k) > n$.

Corollary Since $\binom{n}{k} \cdot 2^{1-\binom{k}{2}} < 1$ for $n = 2^{k/2}$ (that is if $k = 2\lg n$), the above theorem implies that $R(k,k) > 2^{k/2}$ for all $k \geq 3$.
Proof of Theorem: Let n, k satisfy the assumption of the theorem. Consider a random 2-coloring of the edges of K_n by red or blue.

For any fixed set R of k vertices, let A_R be the event that the induced subgraph of K_n on R is **monochromatic** (i.e. that either all its edges are red or they are all blue). Clearly, $Pr(A_R) = 2\frac{1}{2^{\binom{k}{2}}} = 2^{1-\binom{k}{2}}$. Since there are $\binom{n}{k}$ possible choices for

R, the probability that at least one of the events A_R occurs is at most $\binom{n}{k}2^{1-\binom{k}{2}} < 1$.

A LOWER BOUND on RAMSEY NUMBERS

Theorem: If $\binom{n}{k} \cdot 2^{1-\binom{k}{2}} < 1$, then $R(k,k) > n$.

Corollary Since $\binom{n}{k} \cdot 2^{1-\binom{k}{2}} < 1$ for $n = 2^{k/2}$ (that is if $k = 2\lg n$), the above theorem implies that $R(k,k) > 2^{k/2}$ for all $k \geq 3$.
Proof of Theorem: Let n, k satisfy the assumption of the theorem. Consider a random 2-coloring of the edges of K_n by red or blue.

For any fixed set R of k vertices, let A_R be the event that the induced subgraph of K_n on R is **monochromatic** (i.e. that either all its edges are red or they are all blue). Clearly, $Pr(A_R) = 2\frac{1}{2^{\binom{k}{2}}} = 2^{1-\binom{k}{2}}$. Since there are $\binom{n}{k}$ possible choices for

R, the probability that at least one of the events A_R occurs is at most $\binom{n}{k}2^{1-\binom{k}{2}} < 1$. Thus, with positive probability, no event A_R occurs and therefore there is a 2-coloring of K_n without a monochromatic K_k, that is, $R(k,k) > n$.

A LOWER BOUND on RAMSEY NUMBERS

Theorem: If $\binom{n}{k} \cdot 2^{1-\binom{k}{2}} < 1$, then $R(k, k) > n$.

Corollary Since $\binom{n}{k} \cdot 2^{1-\binom{k}{2}} < 1$ for $n = 2^{k/2}$ (that is if $k = 2 \lg n$), the above theorem implies that $R(k, k) > 2^{k/2}$ for all $k \geq 3$.

Proof of Theorem: Let n, k satisfy the assumption of the theorem. Consider a random 2-coloring of the edges of K_n by red or blue.

For any fixed set R of k vertices, let A_R be the event that the induced subgraph of K_n on R is **monochromatic** (i.e. that either all its edges are red or they are all blue). Clearly, $Pr(A_R) = 2\frac{1}{2^{\binom{k}{2}}} = 2^{1-\binom{k}{2}}$. Since there are $\binom{n}{k}$ possible choices for

R, the probability that at least one of the events A_R occurs is at most $\binom{n}{k}2^{1-\binom{k}{2}} < 1$. Thus, with positive probability, no event A_R occurs and therefore there is a 2-coloring of K_n without a monochromatic K_k, that is, $R(k, k) > n$. Note that if $k \geq 3$ and $n = \lfloor 2^{k/2} \rfloor$, then

$$\binom{n}{k} 2^{1-\binom{k}{2}} < \frac{2^{1+k/2}}{k!} \frac{n^k}{2^{k^2/2}} < 1$$

and hence $R(k, k) > 2^{k/2}$ for all $k \geq 3$.

ALGORITHMIC CONSEQUENCES

ALGORITHMIC CONSEQUENCES

Last theorem implies that there is an edge two-coloring of K_n without a monochromatic $K_{2 \lg_2 n}$.

ALGORITHMIC CONSEQUENCES

Last theorem implies that there is an edge two-coloring of K_n without a monochromatic $K_{2 \lg_2 n}$. It is therefore natural to ask whether we can find efficiently such a coloring.

ALGORITHMIC CONSEQUENCES

Last theorem implies that there is an edge two-coloring of K_n without a monochromatic $K_{2 \lg_2 n}$. It is therefore natural to ask whether we can find efficiently such a coloring.

Since there are $2^{\binom{n}{2}}$ possible colorings, an exhaustive search cannot be efficient.

ALGORITHMIC CONSEQUENCES

Last theorem implies that there is an edge two-coloring of K_n without a monochromatic $K_{2 \lg_2 n}$. It is therefore natural to ask whether we can find efficiently such a coloring.

Since there are $2^{\binom{n}{2}}$ possible colorings, an exhaustive search cannot be efficient.

However, a closer look at the proof of the last theorem shows that the proof can be used to produce effectively a coloring that is very likely to be good.

ALGORITHMIC CONSEQUENCES

Last theorem implies that there is an edge two-coloring of K_n without a monochromatic $K_{2\lg_2 n}$. It is therefore natural to ask whether we can find efficiently such a coloring.

Since there are $2^{\binom{n}{2}}$ possible colorings, an exhaustive search cannot be efficient.

However, a closer look at the proof of the last theorem shows that the proof can be used to produce effectively a coloring that is very likely to be good. This is due to the fact for large k if $n = \lfloor 2^{k/2} \rfloor$, then

$$\binom{n}{k} 2^{1-\binom{k}{2}} < \frac{2^{1+\frac{k}{2}}}{k!} \left(\frac{n}{2^{k/2}}\right)^k \leq \frac{2^{1+\frac{k}{2}}}{k!} << 1$$

because $\binom{n}{k} \leq \frac{n^k}{k!}$. Hence, a random coloring of K_n is very likely not to contain a monochromatic $K_{2\lg_2 n}$.

A CONSEQUENCE

As a consequence of previous results, if we need to find a two-coloring of edges of K_{1024} without a monochromatic K_{20} we can simply produce a random two-coloring and then the probability that it contains a monochromatic K_{20} is less than $\frac{2^{11}}{20!}$ what is much, much less than probability of error in any proof that a certain coloring is good.

SOME RAMSEY's NUMBERS

For some Ramsey's numbers see

http://en.wikipedia.org/wiki/Ramsey's_theorem

SOME RAMSEY's NUMBERS

For some Ramsey's numbers see

http://en.wikipedia.org/wiki/Ramsey's_theorem

For example,

SOME RAMSEY's NUMBERS

For some Ramsey's numbers see

http://en.wikipedia.org/wiki/Ramsey's_theorem

For example,

$$R(3,3) = 6, \quad R(4,4) = 18$$

SOME RAMSEY's NUMBERS

For some Ramsey's numbers see

http://en.wikipedia.org/wiki/Ramsey's_theorem

For example,

$$R(3,3) = 6, \quad R(4,4) = 18$$

$$43 \leq R(5,5) \leq 49.$$

THE PARTY PROBLEM

THE PARTY PROBLEM

Ramsey problem is also called **Party problem**.

THE PARTY PROBLEM

Ramsey problem is also called **Party problem**.

Nodes of a K_n graph are seen as a party participants and two of them are connected by a red (blue) edge if they are friends (strangers).

THE PARTY PROBLEM

Ramsey problem is also called **Party problem**.
Nodes of a K_n graph are seen as a party participants and two of them are connected by a red (blue) edge if they are friends (strangers).

Ramsey number $R(k, l)$ is the smallest number n such that at any party of n people there are at least k people that are mutually friends and l people that are mutually strangers.

THE PARTY PROBLEM

Ramsey problem is also called **Party problem**.
Nodes of a K_n graph are seen as a party participants and two of them are connected by a red (blue) edge if they are friends (strangers).

Ramsey number $R(k, l)$ is the smallest number n such that at any party of n people there are at least k people that are mutually friends and l people that are mutually strangers.

In 1993 S. P. Radziszowski and B. D. McKay showed that $R(4, 5) = 25$. They estimate that their computer proof consumed an equivalent of 11 years of computation by a standard desktop computer.

DELETION METHOD

DELETION METHOD

So called **deletion method** can be useful in some cases when it seems to be difficult to apply the probabilistic method directly.

DELETION METHOD

So called **deletion method** can be useful in some cases when it seems to be difficult to apply the probabilistic method directly.

The proof, by the deletion method, that a certain combinatorial object $\mathcal{O}$ exists, consists, conceptually, of two stages:

DELETION METHOD

So called **deletion method** can be useful in some cases when it seems to be difficult to apply the probabilistic method directly.

The proof, by the deletion method, that a certain combinatorial object $\mathcal{O}$ exists, consists, conceptually, of two stages:

- It is first shown that with a positive probability an object $\mathcal{O}'$ exists that is very close, in some sense, to $\mathcal{O}$.

DELETION METHOD

So called **deletion method** can be useful in some cases when it seems to be difficult to apply the probabilistic method directly.

The proof, by the deletion method, that a certain combinatorial object $\mathcal{O}$ exists, consists, conceptually, of two stages:

- It is first shown that with a positive probability an object $\mathcal{O}'$ exists that is very close, in some sense, to $\mathcal{O}$.
- Secondly, $\mathcal{O}'$ is changed, to obtain $\mathcal{O}$, and it is shown that the probability of the existence of $\mathcal{O}$ remains positive.

MIN-MAX TRIANGLE PROBLEM - I.

MIN-MAX TRIANGLE PROBLEM - I.

Let $S = \{p_1, \ldots, p_n\}$ be a set of points located in a unit square of the plane.

MIN-MAX TRIANGLE PROBLEM - I.

Let $S = \{p_1, \ldots, p_n\}$ be a set of points located in a unit square of the plane.

Consider a set T_S of all triangles whose vertices are points of S and let $T_\delta(S)$ be the area of al triangles from T_S the area of which is smaller than δ.

MIN-MAX TRIANGLE PROBLEM - I.

Let $S = \{p_1, \ldots, p_n\}$ be a set of points located in a unit square of the plane.

Consider a set T_S of all triangles whose vertices are points of S and let $T_\delta(S)$ be the area of al triangles from T_S the area of which is smaller than δ.

The following theorem asserts that for any n there is a set S of n points such that $T(S)$ is not too small. Namely, it holds:

MIN-MAX TRIANGLE PROBLEM - I.

Let $S = \{p_1, \ldots, p_n\}$ be a set of points located in a unit square of the plane.

Consider a set T_S of all triangles whose vertices are points of S and let $T_\delta(S)$ be the area of al triangles from T_S the area of which is smaller than δ.

The following theorem asserts that for any n there is a set S of n points such that $T(S)$ is not too small. Namely, it holds:

Theorem For any n there is a set S of n points in a unit square such that $T(S) \geq \frac{1}{100n^2}$.

MIN-MAX TRIANGLE PROBLEM - I.

Let $S = \{p_1, \ldots, p_n\}$ be a set of points located in a unit square of the plane.

Consider a set T_S of all triangles whose vertices are points of S and let $T_\delta(S)$ be the area of al triangles from T_S the area of which is smaller than δ.

The following theorem asserts that for any n there is a set S of n points such that $T(S)$ is not too small. Namely, it holds:

Theorem For any n there is a set S of n points in a unit square such that $T(S) \geq \frac{1}{100n^2}$.

PROOF. BASIC IDEA: $2n$ points are chosen randomly in the unit square.

MIN-MAX TRIANGLE PROBLEM - I.

Let $S = \{p_1, \ldots, p_n\}$ be a set of points located in a unit square of the plane.

Consider a set T_S of all triangles whose vertices are points of S and let $T_\delta(S)$ be the area of al triangles from T_S the area of which is smaller than δ.

The following theorem asserts that for any n there is a set S of n points such that $T(S)$ is not too small. Namely, it holds:

Theorem For any n there is a set S of n points in a unit square such that $T(S) \geq \frac{1}{100n^2}$.

PROOF. BASIC IDEA: $2n$ points are chosen randomly in the unit square. All triangles created by these points are tested and those that are "too small" are eliminated - by deleting one vertex from each of them until only n of nodes are left.

MIN-MAX TRIANGLE PROBLEM - I.

Let $S = \{p_1, \ldots, p_n\}$ be a set of points located in a unit square of the plane.

Consider a set T_S of all triangles whose vertices are points of S and let $T_\delta(S)$ be the area of al triangles from T_S the area of which is smaller than δ.

The following theorem asserts that for any n there is a set S of n points such that $T(S)$ is not too small. Namely, it holds:

Theorem For any n there is a set S of n points in a unit square such that $T(S) \geq \frac{1}{100n^2}$.

PROOF. BASIC IDEA: $2n$ points are chosen randomly in the unit square. All triangles created by these points are tested and those that are "too small" are eliminated - by deleting one vertex from each of them until only n of nodes are left. The idea is that this way we will be left with enough points, and with no too-small-area triangle.

MIN-MAX TRIANGLE PROBLEM - II.

MIN-MAX TRIANGLE PROBLEM - II.

PROOF Choose uniformly $2n$ points in the unit square.

MIN-MAX TRIANGLE PROBLEM - II.

PROOF Choose uniformly $2n$ points in the unit square.

For points p, q, r let $A(p, q, r)$ denote the area of the triangle these points create.

MIN-MAX TRIANGLE PROBLEM - II.

PROOF Choose uniformly $2n$ points in the unit square.

For points p, q, r let $A(p, q, r)$ denote the area of the triangle these points create.

For any real numbers $0 \leq b$ and $\Delta b \leq 1$ it holds

$$\Pr[b \leq ||p - q|| \leq b + \Delta b]$$

MIN-MAX TRIANGLE PROBLEM - II.

PROOF Choose uniformly $2n$ points in the unit square.

For points p, q, r let $A(p, q, r)$ denote the area of the triangle these points create.

For any real numbers $0 \leq b$ and $\Delta b \leq 1$ it holds

$$\Pr[b \leq ||p - q|| \leq b + \Delta b] \leq \pi(b + \Delta b)^2 - \pi b^2 =$$

MIN-MAX TRIANGLE PROBLEM - II.

PROOF Choose uniformly $2n$ points in the unit square.

For points p, q, r let $A(p, q, r)$ denote the area of the triangle these points create.

For any real numbers $0 \leq b$ and $\Delta b \leq 1$ it holds

$$\Pr[b \leq ||p - q|| \leq b + \Delta b] \leq \pi(b + \Delta b)^2 - \pi b^2 = 2\pi b \Delta b + \pi(\Delta b)^2,$$

where $||p - q||$ denote the Euclidean distance between p and q.

MIN-MAX TRIANGLE PROBLEM - II.

PROOF Choose uniformly $2n$ points in the unit square.

For points p, q, r let $A(p, q, r)$ denote the area of the triangle these points create.

For any real numbers $0 \leq b$ and $\Delta b \leq 1$ it holds

$$\Pr[b \leq ||p - q|| \leq b + \Delta b] \leq \pi(b + \Delta b)^2 - \pi b^2 = 2\pi b \Delta b + \pi(\Delta b)^2,$$

where $||p - q||$ denote the Euclidean distance between p and q.

(Observe that inequality follows from the fact that rings with radiuses b and $b + \Delta b$ and centre in p may not be completely contained in the unit square.)

MIN-MAX TRIANGLE PROBLEM - II.

PROOF Choose uniformly $2n$ points in the unit square.

For points p, q, r let $A(p, q, r)$ denote the area of the triangle these points create.

For any real numbers $0 \leq b$ and $\Delta b \leq 1$ it holds

$$\Pr[b \leq ||p - q|| \leq b + \Delta b] \leq \pi(b + \Delta b)^2 - \pi b^2 = 2\pi b \Delta b + \pi(\Delta b)^2,$$

where $||p - q||$ denote the Euclidean distance between p and q.

(Observe that inequality follows from the fact that rings with radiuses b and $b + \Delta b$ and centre in p may not be completely contained in the unit square.)

Let (p, q) be the base of the triangle (p, q, r) and let $||p - q|| = b$.

MIN-MAX TRIANGLE PROBLEM - III.

MIN-MAX TRIANGLE PROBLEM - III.

We show now how to estimate $\Pr[A(p, q, r) \leq \varepsilon]$ for any $\varepsilon > 0$ and $p.q.r$.

MIN-MAX TRIANGLE PROBLEM - III.

We show now how to estimate $\Pr[A(p,q,r) \leq \varepsilon]$ for any $\varepsilon > 0$ and $p.q.r$.

Such an event happens when the height h of the triangle is $\leq \frac{2\varepsilon}{b}$ and therefore r is not farther than $\frac{2\varepsilon}{b}$ from the line of points p and q.

MIN-MAX TRIANGLE PROBLEM - III.

We show now how to estimate $\Pr[A(p,q,r) \leq \varepsilon]$ for any $\varepsilon > 0$ and $p.q.r$.

Such an event happens when the height h of the triangle is $\leq \frac{2\varepsilon}{b}$ and therefore r is not farther than $\frac{2\varepsilon}{b}$ from the line of points p and q.

The probability that this happens is less than $\frac{2\varepsilon\sqrt{2}}{b}$, because r has to be in a strip of width $\frac{2\varepsilon}{b}$ and length less than $\sqrt{2}$. Hence,

MIN-MAX TRIANGLE PROBLEM - III.

We show now how to estimate $\Pr[A(p,q,r) \leq \varepsilon]$ for any $\varepsilon > 0$ and $p.q.r$.

Such an event happens when the height h of the triangle is $\leq \frac{2\varepsilon}{b}$ and therefore r is not farther than $\frac{2\varepsilon}{b}$ from the line of points p and q.

The probability that this happens is less than $\frac{2\varepsilon\sqrt{2}}{b}$, because r has to be in a strip of width $\frac{2\varepsilon}{b}$ and length less than $\sqrt{2}$. Hence,

$$\begin{aligned}\Pr[A(p,q,r) \leq \varepsilon] &= \int_{b=0}^{\sqrt{2}} \Pr[b \leq ||p-q|| \leq b + \Delta b] \times \Pr[\text{triangle. h.} \leq \frac{2\varepsilon}{b}] \\ &\leq \int_{b=0}^{\sqrt{2}} \frac{2\sqrt{2}\varepsilon}{b} 4\pi b \Delta b = 16\pi\varepsilon.\end{aligned}$$

MIN-MAX TRIANGLE PROBLEM - IV

Let us now compute the expected number of triangles with the area $\leq \varepsilon = \frac{1}{100n^2}$.

Let S' be a set of $2n$ points uniformly distributed in the unit square. For each triple (p_i, q_i, r_i) in S' let X_{p_i,q_i,r_i} be the indicator variable having value 1 if the area of the triangle determined by (p_i, q_i, r_i) is less than $\varepsilon = \frac{1}{100n^2}$.

The probability that the area of some specific triangle is less than $\frac{1}{100n^2}$ is less than

$$16\pi\varepsilon = \frac{16\pi}{100n^2} \leq \frac{0.6}{n^2}$$

This is also the expected value of X_{p_i,q_i,r_i}.

MIN-MAX TRIANGLE PROBLEM - V.

If X denotes the number of triangles with area less than $\frac{1}{100n^2}$, then

$$\mathbf{E}[X] = \sum_{p,q,r \in S'} \mathbf{E}[X_{p_i,q_i,r_i}] \leq \binom{2n}{3} 0.6n^{-2} \leq n.$$

Finally, by throwing away an arbitrary vertex from each of such "small area triangles", we are left with a new set S'' of points the expected size of which (of S''), is not less than n, in which no small-area-triangles exist.

MIN-MAX TRIANGLE PROBLEM - V.

If X denotes the number of triangles with area less than $\frac{1}{100n^2}$, then

$$\mathbf{E}[X] = \sum_{p,q,r \in S'} \mathbf{E}[X_{p_i,q_i,r_i}] \leq \binom{2n}{3} 0.6n^{-2} \leq n.$$

Finally, by throwing away an arbitrary vertex from each of such "small area triangles", we are left with a new set S'' of points the expected size of which (of S''), is not less than n, in which no small-area-triangles exist.

Therefore, there exists a set S'', of size n, such that $T(S'') \geq \frac{1}{100n^2}$.

EXAMPLE - INDEPENDENT SETS OF VERTICES

EXAMPLE - INDEPENDENT SETS OF VERTICES

We show a lower bound on the size of the largest independent set of vertices in certain graphs.

EXAMPLE - INDEPENDENT SETS OF VERTICES

We show a lower bound on the size of the largest independent set of vertices in certain graphs.

Definition An independent set of a graph $G = (V, E)$ is a subset of vertices of V such that no two vertices in the set are adjacent. Denote by $\alpha(G)$ the size of the largest independent set of vertices of the graph G.

EXAMPLE - INDEPENDENT SETS OF VERTICES

We show a lower bound on the size of the largest independent set of vertices in certain graphs.

Definition An independent set of a graph $G = (V, E)$ is a subset of vertices of V such that no two vertices in the set are adjacent. Denote by $\alpha(G)$ the size of the largest independent set of vertices of the graph G.

Our aim is to prove the following lower bound (for any integer k):

EXAMPLE - INDEPENDENT SETS OF VERTICES

We show a lower bound on the size of the largest independent set of vertices in certain graphs.

Definition An independent set of a graph $G = (V, E)$ is a subset of vertices of V such that no two vertices in the set are adjacent. Denote by $\alpha(G)$ the size of the largest independent set of vertices of the graph G.

Our aim is to prove the following lower bound (for any integer k):

Theorem If $|V| = n$ and $|E| = nk/2$ for a graph $G = (V, E)$, then $\alpha(G) \geq \frac{n}{2k}$.

EXAMPLE - INDEPENDENT SETS OF VERTICES

We show a lower bound on the size of the largest independent set of vertices in certain graphs.

Definition An independent set of a graph $G = (V, E)$ is a subset of vertices of V such that no two vertices in the set are adjacent. Denote by $\alpha(G)$ the size of the largest independent set of vertices of the graph G.

Our aim is to prove the following lower bound (for any integer k):

Theorem If $|V| = n$ and $|E| = nk/2$ for a graph $G = (V, E)$, then $\alpha(G) \geq \frac{n}{2k}$.

Proof ideas:

EXAMPLE - INDEPENDENT SETS OF VERTICES

We show a lower bound on the size of the largest independent set of vertices in certain graphs.

Definition An independent set of a graph $G = (V, E)$ is a subset of vertices of V such that no two vertices in the set are adjacent. Denote by $\alpha(G)$ the size of the largest independent set of vertices of the graph G.

Our aim is to prove the following lower bound (for any integer k):

Theorem If $|V| = n$ and $|E| = nk/2$ for a graph $G = (V, E)$, then $\alpha(G) \geq \frac{n}{2k}$.

Proof ideas:

- To choose randomly a subset of vertices that would be a candidate for an independent set.

EXAMPLE - INDEPENDENT SETS OF VERTICES

We show a lower bound on the size of the largest independent set of vertices in certain graphs.

Definition An independent set of a graph $G = (V, E)$ is a subset of vertices of V such that no two vertices in the set are adjacent. Denote by $\alpha(G)$ the size of the largest independent set of vertices of the graph G.

Our aim is to prove the following lower bound (for any integer k):

Theorem If $|V| = n$ and $|E| = nk/2$ for a graph $G = (V, E)$, then $\alpha(G) \geq \frac{n}{2k}$.

Proof ideas:

- To choose randomly a subset of vertices that would be a candidate for an independent set.
- To show, using the probabilistic argument, that there is a subset of the chosen set of vertices that has many more vertices than edges.

EXAMPLE - INDEPENDENT SETS OF VERTICES

We show a lower bound on the size of the largest independent set of vertices in certain graphs.

Definition An independent set of a graph $G = (V, E)$ is a subset of vertices of V such that no two vertices in the set are adjacent. Denote by $\alpha(G)$ the size of the largest independent set of vertices of the graph G.

Our aim is to prove the following lower bound (for any integer k):

Theorem If $|V| = n$ and $|E| = nk/2$ for a graph $G = (V, E)$, then $\alpha(G) \geq \frac{n}{2k}$.

Proof ideas:

- To choose randomly a subset of vertices that would be a candidate for an independent set.
- To show, using the probabilistic argument, that there is a subset of the chosen set of vertices that has many more vertices than edges.
- By deleting one vertex from each of such edges, an independent set is produced.

EXAMPLE - INDEPENDENT SETS OF VERTICES

We show a lower bound on the size of the largest independent set of vertices in certain graphs.

Definition An independent set of a graph $G = (V, E)$ is a subset of vertices of V such that no two vertices in the set are adjacent. Denote by $\alpha(G)$ the size of the largest independent set of vertices of the graph G.

Our aim is to prove the following lower bound (for any integer k):

Theorem If $|V| = n$ and $|E| = nk/2$ for a graph $G = (V, E)$, then $\alpha(G) \geq \frac{n}{2k}$.

Proof ideas:

- To choose randomly a subset of vertices that would be a candidate for an independent set.
- To show, using the probabilistic argument, that there is a subset of the chosen set of vertices that has many more vertices than edges.
- By deleting one vertex from each of such edges, an independent set is produced.

PROOF

PROOF

Create a set $S \subset V$ by putting into S each vertex independently with probability p (to be specified later).

PROOF

Create a set $S \subset V$ by putting into S each vertex independently with probability p (to be specified later). It therefore holds for the average size of S that:
$\mathbf{E}[|S|] = np$.

PROOF

Create a set $S \subset V$ by putting into S each vertex independently with probability p (to be specified later). It therefore holds for the average size of S that: $\mathbf{E}[|S|] = np$. Let G_S be the subgraph of $G = (V, E)$, induced by S.

PROOF

Create a set $S \subset V$ by putting into S each vertex independently with probability p (to be specified later). It therefore holds for the average size of S that: $\mathbf{E}[|S|] = np$. Let G_S be the subgraph of $G = (V, E)$, induced by S.

For any $e \in E$ let Y_e be the indicator variable that has value 1 if $e \in E(G_S)$ - the set of edges of G_S - and 0 otherwise.

PROOF

Create a set $S \subset V$ by putting into S each vertex independently with probability p (to be specified later). It therefore holds for the average size of S that: $\mathbf{E}[|S|] = np$. Let G_S be the subgraph of $G = (V, E)$, induced by S.

For any $e \in E$ let Y_e be the indicator variable that has value 1 if $e \in E(G_S)$ - the set of edges of G_S - and 0 otherwise.

$\mathbf{E}[Y_e] = p^2$ because an edge belongs to $E(G_S)$ iff both if its endpoints are in S, what happens with probability p^2.

PROOF

Create a set $S \subset V$ by putting into S each vertex independently with probability p (to be specified later). It therefore holds for the average size of S that: $\mathbf{E}[|S|] = np$. Let G_S be the subgraph of $G = (V, E)$, induced by S.

For any $e \in E$ let Y_e be the indicator variable that has value 1 if $e \in E(G_S)$ - the set of edges of G_S - and 0 otherwise.

$\mathbf{E}[Y_e] = p^2$ because an edge belongs to $E(G_S)$ iff both if its endpoints are in S, what happens with probability p^2. Let $Y = |E(G_S)|$. It holds

$$\mathbf{E}[Y] = \mathbf{E}[\sum_{e \in E} Y_e] = \sum_{e \in E} \mathbf{E}[Y_e] = \frac{nk}{2} p^2.$$

PROOF

Create a set $S \subset V$ by putting into S each vertex independently with probability p (to be specified later). It therefore holds for the average size of S that: $\mathbf{E}[|S|] = np$. Let G_S be the subgraph of $G = (V, E)$, induced by S.

For any $e \in E$ let Y_e be the indicator variable that has value 1 if $e \in E(G_S)$ - the set of edges of G_S - and 0 otherwise.

$\mathbf{E}[Y_e] = p^2$ because an edge belongs to $E(G_S)$ iff both if its endpoints are in S, what happens with probability p^2. Let $Y = |E(G_S)|$. It holds

$$\mathbf{E}[Y] = \mathbf{E}[\sum_{e \in E} Y_e] = \sum_{e \in E} \mathbf{E}[Y_e] = \frac{nk}{2} p^2.$$

After deleting all edges from G_S, by dropping a vertex from each of such edges, it remains a set S^* of the expected size $\mathbf{E}[|S^*|] = \mathbf{E}[|S| - Y]$, and therefore

$$\mathbf{E}[|S| - Y] = \mathbf{E}[|S|] - \mathbf{E}[Y] = np - \frac{nk}{2} p^2.$$

PROOF

Create a set $S \subset V$ by putting into S each vertex independently with probability p (to be specified later). It therefore holds for the average size of S that: $\mathbf{E}[|S|] = np$. Let G_S be the subgraph of $G = (V, E)$, induced by S.

For any $e \in E$ let Y_e be the indicator variable that has value 1 if $e \in E(G_S)$ - the set of edges of G_S - and 0 otherwise.

$\mathbf{E}[Y_e] = p^2$ because an edge belongs to $E(G_S)$ iff both if its endpoints are in S, what happens with probability p^2. Let $Y = |E(G_S)|$. It holds

$$\mathbf{E}[Y] = \mathbf{E}[\sum_{e \in E} Y_e] = \sum_{e \in E} \mathbf{E}[Y_e] = \frac{nk}{2} p^2.$$

After deleting all edges from G_S, by dropping a vertex from each of such edges, it remains a set S^* of the expected size $\mathbf{E}[|S^*|] = \mathbf{E}[|S| - Y]$, and therefore

$$\mathbf{E}[|S| - Y] = \mathbf{E}[|S|] - \mathbf{E}[Y] = np - \frac{nk}{2} p^2.$$

The last expression has the largest value for $p = \frac{1}{k}$ and in such a case

$$\mathbf{E}[|S^*|] = \mathbf{E}[|S| - Y] = \frac{n}{2k}.$$

EXPANDING GRAPHS

EXPANDING GRAPHS

Informally, an **expanding graph** is a graph in which the number of neighbors of any sufficiently small set of vertices S is larger than $c|S|$ for some positive constant $c > 1$.

EXPANDING GRAPHS

Informally, an **expanding graph** is a graph in which the number of neighbors of any sufficiently small set of vertices S is larger than $c|S|$ for some positive constant $c > 1$.

OR-concentrators are a special type of expanding graphs.

EXPANDING GRAPHS

Informally, an **expanding graph** is a graph in which the number of neighbors of any sufficiently small set of vertices S is larger than $c|S|$ for some positive constant $c > 1$.

OR-concentrators are a special type of expanding graphs.

Definition: An (n, d, α, c) OR-concentrator is a bipartite multigraph $G = (L, R, E)$ with independent sets of vertices L and R, each of cardinality n, such that

1. Every vertex in L has degree at most d;
2. For any subset S of vertices from L such that $|S| \leq \alpha n$ there are more than $c|S|$ neighbors in R.

EXPANDING GRAPHS

Informally, an **expanding graph** is a graph in which the number of neighbors of any sufficiently small set of vertices S is larger than $c|S|$ for some positive constant $c > 1$.

OR-concentrators are a special type of expanding graphs.

Definition: An (n, d, α, c) OR-concentrator is a bipartite multigraph $G = (L, R, E)$ with independent sets of vertices L and R, each of cardinality n, such that

1. Every vertex in L has degree at most d;
2. For any subset S of vertices from L such that $|S| \leq \alpha n$ there are more than $c|S|$ neighbors in R.

For applications, it is usually desirable to have d as small and c as large as possible.

EXPANDING GRAPHS

Informally, an **expanding graph** is a graph in which the number of neighbors of any sufficiently small set of vertices S is larger than $c|S|$ for some positive constant $c > 1$.

OR-concentrators are a special type of expanding graphs.

Definition: An (n, d, α, c) OR-concentrator is a bipartite multigraph $G = (L, R, E)$ with independent sets of vertices L and R, each of cardinality n, such that

1. Every vertex in L has degree at most d;
2. For any subset S of vertices from L such that $|S| \leq \alpha n$ there are more than $c|S|$ neighbors in R.

For applications, it is usually desirable to have d as small and c as large as possible.

Of particular interest is to study OR-concentrators in which α, c and d are constants fixed independently of n, with $c > 1$.

EXPANDING GRAPHS

Informally, an **expanding graph** is a graph in which the number of neighbors of any sufficiently small set of vertices S is larger than $c|S|$ for some positive constant $c > 1$.

OR-concentrators are a special type of expanding graphs.

Definition: An (n, d, α, c) OR-concentrator is a bipartite multigraph $G = (L, R, E)$ with independent sets of vertices L and R, each of cardinality n, such that

1. Every vertex in L has degree at most d;
2. For any subset S of vertices from L such that $|S| \leq \alpha n$ there are more than $c|S|$ neighbors in R.

For applications, it is usually desirable to have d as small and c as large as possible.

Of particular interest is to study OR-concentrators in which α, c and d are constants fixed independently of n, with $c > 1$.

Finding an explicit construction of OR-concentrators is a non-trivial task. However, the probabilistic method can be used to show the existence of such concentrators.

Theorem: There is an integer n_0 such that for all $n > n_0$ there is an $(n, 18, \frac{1}{3}, 2)$ OR-concentrator.

Theorem: There is an integer n_0 such that for all $n > n_0$ there is an $(n, 18, \frac{1}{3}, 2)$ OR-concentrator.

Proof: The first part of the proof will be for all (n, d, c, α) OR-concentrators.

Theorem: There is an integer n_0 such that for all $n > n_0$ there is an $(n, 18, \frac{1}{3}, 2)$ OR-concentrator.

Proof: The first part of the proof will be for all (n, d, c, α) OR-concentrators.

Consider a random bipartite graph with two disjoint sets of vertices, L and R, each of n vertices, in which each vertex of L chooses randomly and independently d vertices from R as neighbours.

Theorem: There is an integer n_0 such that for all $n > n_0$ there is an $(n, 18, \frac{1}{3}, 2)$ OR-concentrator.

Proof: The first part of the proof will be for all (n, d, c, α) OR-concentrators.

Consider a random bipartite graph with two disjoint sets of vertices, L and R, each of n vertices, in which each vertex of L chooses randomly and independently d vertices from R as neighbours.

For any integer s **let ξ_s denote the event that a (bad) subset of s vertices of L has fewer than cs neighbours in R.**

Theorem: There is an integer n_0 such that for all $n > n_0$ there is an $(n, 18, \frac{1}{3}, 2)$ OR-concentrator.

Proof: The first part of the proof will be for all (n, d, c, α) OR-concentrators.

Consider a random bipartite graph with two disjoint sets of vertices, L and R, each of n vertices, in which each vertex of L chooses randomly and independently d vertices from R as neighbours.

For any integer s **let ξ_s denote the event that a (bad) subset of s vertices of L has fewer than cs neighbours in R.**

We first derive an upper bound on $Pr[\xi_s]$, for any particular fixed s, and then we show the upper bound on the sum of $\Pr[\xi_s]$ over all $s \leq \alpha n$.

Theorem: There is an integer n_0 such that for all $n > n_0$ there is an $(n, 18, \frac{1}{3}, 2)$ OR-concentrator.

Proof: The first part of the proof will be for all (n, d, c, α) OR-concentrators.

Consider a random bipartite graph with two disjoint sets of vertices, L and R, each of n vertices, in which each vertex of L chooses randomly and independently d vertices from R as neighbours.

For any integer s **let ξ_s denote the event that a (bad) subset of s vertices of L has fewer than cs neighbours in R.**

We first derive an upper bound on $Pr[\xi_s]$, for any particular fixed s, and then we show the upper bound on the sum of $\Pr[\xi_s]$ over all $s \leq \alpha n$. **This way we obtain a non-trivial upper bound on the probability that a random graph with parameters we seek, fails to be an OR-concentrator.**

Theorem: There is an integer n_0 such that for all $n > n_0$ there is an $(n, 18, \frac{1}{3}, 2)$ OR-concentrator.

Proof: The first part of the proof will be for all (n, d, c, α) OR-concentrators.

Consider a random bipartite graph with two disjoint sets of vertices, L and R, each of n vertices, in which each vertex of L chooses randomly and independently d vertices from R as neighbours.

For any integer s **let ξ_s denote the event that a (bad) subset of s vertices of L has fewer than cs neighbours in R.**

We first derive an upper bound on $Pr[\xi_s]$, for any particular fixed s, and then we show the upper bound on the sum of $\Pr[\xi_s]$ over all $s \leq \alpha n$. **This way we obtain a non-trivial upper bound on the probability that a random graph with parameters we seek, fails to be an OR-concentrator.**

Fix any subset $S \subseteq L$ of size s, and any subset $T \subseteq R$ of size cs.

(There are $\binom{n}{s}$ ways of choosing S, and $\binom{n}{cs}$ ways of choosing T.)

Theorem: There is an integer n_0 such that for all $n > n_0$ there is an $(n, 18, \frac{1}{3}, 2)$ OR-concentrator.

Proof: The first part of the proof will be for all (n, d, c, α) OR-concentrators.

Consider a random bipartite graph with two disjoint sets of vertices, L and R, each of n vertices, in which each vertex of L chooses randomly and independently d vertices from R as neighbours.

For any integer s **let ξ_s denote the event that a (bad) subset of s vertices of L has fewer than cs neighbours in R.**

We first derive an upper bound on $Pr[\xi_s]$, for any particular fixed s, and then we show the upper bound on the sum of $\Pr[\xi_s]$ over all $s \leq \alpha n$. **This way we obtain a non-trivial upper bound on the probability that a random graph with parameters we seek, fails to be an OR-concentrator.**

Fix any subset $S \subseteq L$ of size s, and any subset $T \subseteq R$ of size cs.

(There are $\binom{n}{s}$ ways of choosing S, and $\binom{n}{cs}$ ways of choosing T.)

The probability that T contains all of at most ds neighbours of the vertices in S is $\left(\frac{cs}{n}\right)^{ds}$.

The probability of the event that all ds edges going out from some s vertices of L fall within any cs vertices of R is bounded by

$$\Pr[\xi_s] \leq \binom{n}{s}\binom{n}{cs}\left(\frac{cs}{n}\right)^{ds}.$$

The probability of the event that all ds edges going out from some s vertices of L fall within any cs vertices of R is bounded by

$$\Pr[\xi_s] \leq \binom{n}{s}\binom{n}{cs}\left(\frac{cs}{n}\right)^{ds}.$$

Using the inequality $\binom{n}{s} \leq \left(\frac{ne}{s}\right)^s$ we obtain

$$\begin{aligned}\Pr[\xi_s] &\leq \binom{n}{s}\binom{n}{cs}\left(\frac{cs}{n}\right)^{ds} \\ &\leq \end{aligned}$$

The probability of the event that all ds edges going out from some s vertices of L fall within any cs vertices of R is bounded by

$$\Pr[\xi_s] \leq \binom{n}{s}\binom{n}{cs}\left(\frac{cs}{n}\right)^{ds}.$$

Using the inequality $\binom{n}{s} \leq \left(\frac{ne}{s}\right)^s$ we obtain

$$\begin{aligned}\Pr[\xi_s] &\leq \binom{n}{s}\binom{n}{cs}\left(\frac{cs}{n}\right)^{ds} \\ &\leq \left(\frac{ne}{s}\right)^s\left(\frac{ne}{cs}\right)^{cs}\left(\frac{cs}{n}\right)^{ds} \\ &=\end{aligned}$$

The probability of the event that all ds edges going out from some s vertices of L fall within any cs vertices of R is bounded by

$$\Pr[\xi_s] \leq \binom{n}{s} \binom{n}{cs} \left(\frac{cs}{n}\right)^{ds}.$$

Using the inequality $\binom{n}{s} \leq \left(\frac{ne}{s}\right)^s$ we obtain

$$\begin{aligned} \Pr[\xi_s] &\leq \binom{n}{s} \binom{n}{cs} \left(\frac{cs}{n}\right)^{ds} \\ &\leq \left(\frac{ne}{s}\right)^{s} \left(\frac{ne}{cs}\right)^{cs} \left(\frac{cs}{n}\right)^{ds} \\ &= \left[\left(\frac{s}{n}\right)^{d-c-1} e^{1+c}\, c^{d-c}\right]^{s}. \end{aligned}$$

In a special case, for $\alpha = \frac{1}{3}$ and $s \leq \alpha n$ we have

In a special case, for $\alpha = \frac{1}{3}$ and $s \leq \alpha n$ we have

$$\Pr[\xi_s] = \left[\left(\frac{1}{3}\right)^{d-c-1} e^{1+c}\, c^{d-c}\right]^s$$

$$\leq$$

In a special case, for $\alpha = \frac{1}{3}$ and $s \leq \alpha n$ we have

$$\Pr[\xi_s] = \left[\left(\frac{1}{3}\right)^{d-c-1} e^{1+c}\, c^{d-c}\right]^s$$

$$\leq \left[\left(\frac{c}{3}\right)^{d} (3e)^{c+1}\right]^s,$$

In a special case, for $\alpha = \frac{1}{3}$ and $s \leq \alpha n$ we have

$$\begin{aligned} \Pr[\xi_s] &= \left[\left(\frac{1}{3}\right)^{d-c-1} e^{1+c}\, c^{d-c}\right]^s \\ &\leq \left[\left(\frac{c}{3}\right)^{d} (3e)^{c+1}\right]^s, \end{aligned}$$

and, in addition, for $c = 2,\ d = 18$

$$\Pr[\xi_s] = \left[\left(\frac{2}{3}\right)^{18} (3e)^{3}\right]^s.$$

In a special case, for $\alpha = \frac{1}{3}$ and $s \leq \alpha n$ we have

$$\begin{aligned}\Pr[\xi_s] &= \left[\left(\frac{1}{3}\right)^{d-c-1} e^{1+c}\, c^{d-c}\right]^s \\ &\leq \left[\left(\frac{c}{3}\right)^d (3e)^{c+1}\right]^s,\end{aligned}$$

and, in addition, for $c = 2,\ d = 18$

$$\Pr[\xi_s] = \left[\left(\frac{2}{3}\right)^{18} (3e)^3\right]^s.$$

Since:

$$r = \left(\frac{2}{3}\right)^{18} (3e)^3 \leq \frac{1}{2}$$

we have

$$\sum_{s>1} Pr[\xi_s] \leq \sum_{s>1} r^s = \frac{r}{1-r} < 1.$$

CONSEQUENCE

The probability that there exists an $(n, 18, \frac{1}{3}, 2)$ concentrator is therefore positive.

RANDOMIZED PERMUTATION ROUTING on HYPERCUBES

The first result concerning permutation routing from the previous chapter said that some oblivious routings, for example the so-called left to right routing, are very simple, but they may take expenential time for the delivery of some permutations.

RANDOMIZED PERMUTATION ROUTING on HYPERCUBES

The first result concerning permutation routing from the previous chapter said that some oblivious routings, for example the so-called left to right routing, are very simple, but they may take expenential time for the delivery of some permutations.

The second result said that randomized oblivious routing algorithms cn be much more efficient concerning the number of steps. Namely, that there is a randomized oblivious routing algorithm that can route any permutation in 15d steps with probability $1 - \frac{1}{n}$.

OBLIVIOUS ROUTING REVISITED - I.

OBLIVIOUS ROUTING REVISITED - I.

As another example, we will show that probabilistic method can be used to prove the existence of a routing algorithm that has as good performance, as the previous one, concerning the number of routing steps and uses much less randomness to do that.

OBLIVIOUS ROUTING REVISITED - I.

As another example, we will show that probabilistic method can be used to prove the existence of a routing algorithm that has as good performance, as the previous one, concerning the number of routing steps and uses much less randomness to do that.

We focus on the minimal number of random bits needed by randomized oblivious routing algorithms for hypercubes and we derive the following results.

OBLIVIOUS ROUTING REVISITED - I.

As another example, we will show that probabilistic method can be used to prove the existence of a routing algorithm that has as good performance, as the previous one, concerning the number of routing steps and uses much less randomness to do that.

We focus on the minimal number of random bits needed by randomized oblivious routing algorithms for hypercubes and we derive the following results.

1. **A proof of the existence (by the probabilistic method) of a randomized routing algorithm that uses (within a constant factor) only $3d$ random bits to route d-dimensional hypercubes.**

OBLIVIOUS ROUTING REVISITED - I.

As another example, we will show that probabilistic method can be used to prove the existence of a routing algorithm that has as good performance, as the previous one, concerning the number of routing steps and uses much less randomness to do that.

We focus on the minimal number of random bits needed by randomized oblivious routing algorithms for hypercubes and we derive the following results.

1. **A proof of the existence (by the probabilistic method) of a randomized routing algorithm that uses (within a constant factor) only $3d$ random bits to route d-dimensional hypercubes.**

   As a consequence we get that our randomized oblivious routing algorithm, from previous chapter, that used $d2^d$ random bits to route a d-dimensional hypercube, uses much too much random bits.

OBLIVIOUS ROUTING REVISITED - II.

OBLIVIOUS ROUTING REVISITED - II.

To remember: $\sqrt{\frac{2^d}{d}}$ is a lower bound on oblivious deterministic routings on a d-dimensional hypercube H_d.

OBLIVIOUS ROUTING REVISITED - II.

To remember: $\sqrt{\frac{2^d}{d}}$ is a lower bound on oblivious deterministic routings on a d-dimensional hypercube H_d.

Question: *How much randomness (how many random bits) is (are) needed to have a routing algorithm with the expected running time $O(d)$?*

OBLIVIOUS ROUTING REVISITED - II.

To remember: $\sqrt{\frac{2^d}{d}}$ is a lower bound on oblivious deterministic routings on a d-dimensional hypercube H_d.

Question: *How much randomness (how many random bits) is (are) needed to have a routing algorithm with the expected running time $O(d)$?*

Observation I: A randomized oblivious algorithm for permutation routing is a probability distribution on a set of deterministic oblivious routing algorithms.

OBLIVIOUS ROUTING REVISITED - II.

To remember: $\sqrt{\frac{2^d}{d}}$ is a lower bound on oblivious deterministic routings on a d-dimensional hypercube H_d.

Question: *How much randomness (how many random bits) is (are) needed to have a routing algorithm with the expected running time $O(d)$?*

Observation I: A randomized oblivious algorithm for permutation routing is a probability distribution on a set of deterministic oblivious routing algorithms.

Observation II. Each deterministic oblivious algorithm for a 2^d-node network is a set of $2^{2d} = 2^d \times 2^d$ routes, one for each source-target pair.

OBLIVIOUS ROUTING REVISITED - II.

To remember: $\sqrt{\frac{2^d}{d}}$ is a lower bound on oblivious deterministic routings on a d-dimensional hypercube H_d.

Question: *How much randomness (how many random bits) is (are) needed to have a routing algorithm with the expected running time $O(d)$?*

Observation I: A randomized oblivious algorithm for permutation routing is a probability distribution on a set of deterministic oblivious routing algorithms.

Observation II. Each deterministic oblivious algorithm for a 2^d-node network is a set of $2^{2d} = 2^d \times 2^d$ routes, one for each source-target pair.

Note: Every randomized oblivious algorithm can be expressed by sequences

$$(A_1, \ldots, A_r), \quad (p_1, \ldots, p_r),$$

where each A_j is a deterministic oblivious routing algorithm and each p_j is the probability that we use A_j on a run of the randomized routing algorithm.

Note: We know:

Note: We know:

1. With 0 random bits the expected running time of any oblivious routing algorithm on H_d is $\Omega\left(\sqrt{\frac{2^d}{d}}\right)$;

Note: We know:

1. With 0 random bits the expected running time of any oblivious routing algorithm on H_d is $\Omega\left(\sqrt{\frac{2^d}{d}}\right)$;
2. For the randomized oblivious routing the expected running time is $O(d)$ and $d\,2^d$ random bits are used (each of 2^d nodes chooses a random d-bit auxiliary goal).

Note: We know:

1. With 0 random bits the expected running time of any oblivious routing algorithm on H_d is $\Omega\left(\sqrt{\frac{2^d}{d}}\right)$;
2. For the randomized oblivious routing the expected running time is $O(d)$ and $d\,2^d$ random bits are used (each of 2^d nodes chooses a random d-bit auxiliary goal).

Are so many random bits indeed necessary for efficient randomized routing?

Theorem: For every d there exists a randomized oblivious scheme (algorithm) for a permutation routing on the hypercube with $n = 2^d$ nodes that uses only $3d$ random bits and still runs in the expected time $15d$ at most.

Theorem: For every d there exists a randomized oblivious scheme (algorithm) for a permutation routing on the hypercube with $n = 2^d$ nodes that uses only $3d$ random bits and still runs in the expected time $15d$ at most.

Proof: Notation We say that a set $\mathcal{B} = \{B_1, \ldots, B_t\}$ of deterministic oblivious permutation routing algorithms on H_d is an efficient routing scheme, if for any input instance, the expected number of steps using a randomly chosen algorithm from $\mathcal{B}$ is at most $15d$.

Theorem: For every d there exists a randomized oblivious scheme (algorithm) for a permutation routing on the hypercube with $n = 2^d$ nodes that uses only $3d$ random bits and still runs in the expected time $15d$ at most.

Proof: Notation We say that a set $\mathcal{B} = \{B_1, \ldots, B_t\}$ of deterministic oblivious permutation routing algorithms on H_d is an efficient routing scheme, if for any input instance, the expected number of steps using a randomly chosen algorithm from $\mathcal{B}$ is at most $15d$.

To prove the theorem, we show that for every $n = 2^d$ there is an efficient routing scheme for H_d with $t = 2^{3d} = n^3$.

Theorem: For every d there exists a randomized oblivious scheme (algorithm) for a permutation routing on the hypercube with $n = 2^d$ nodes that uses only $3d$ random bits and still runs in the expected time $15d$ at most.

Proof: Notation We say that a set $\mathcal{B} = \{B_1, \ldots, B_t\}$ of deterministic oblivious permutation routing algorithms on H_d is an efficient routing scheme, if for any input instance, the expected number of steps using a randomly chosen algorithm from $\mathcal{B}$ is at most $15d$.

To prove the theorem, we show that for every $n = 2^d$ there is an efficient routing scheme for H_d with $t = 2^{3d} = n^3$.

Our resulting randomized routing scheme will randomly choose n^3 of n^n possible deterministic oblivious routing algorithms. (n^n is due to the fact that there are n sources and for each one we can choose from n possible intermediate destinations.)

Theorem: For every d there exists a randomized oblivious scheme (algorithm) for a permutation routing on the hypercube with $n = 2^d$ nodes that uses only $3d$ random bits and still runs in the expected time $15d$ at most.

Proof: Notation We say that a set $\mathcal{B} = \{B_1, \ldots, B_t\}$ of deterministic oblivious permutation routing algorithms on H_d is an efficient routing scheme, if for any input instance, the expected number of steps using a randomly chosen algorithm from $\mathcal{B}$ is at most $15d$.

To prove the theorem, we show that for every $n = 2^d$ there is an efficient routing scheme for H_d with $t = 2^{3d} = n^3$.

Our resulting randomized routing scheme will randomly choose n^3 of n^n possible deterministic oblivious routing algorithms. (n^n is due to the fact that there are n sources and for each one we can choose from n possible intermediate destinations.)

Let us denote such deterministic algorithms by $A_j,\ 1 \leq j \leq n^n$. On an n-node network there are $n!$ distinct possible instances of the permutation routing problem, one for each permutation on $\{1, 2, \ldots, n\}$.

For a permutation $\pi_i, \ 1 \leq i \leq n!$, let us call a deterministic oblivious routing algorithm A_j **good** if A_j routes π_i in $14d$ or fewer steps, and **bad** otherwise.

For a permutation π_i, $1 \leq i \leq n!$, let us call a deterministic oblivious routing algorithm A_j **good** if A_j routes π_i in $14d$ or fewer steps, and **bad** otherwise.

By our randomized routing result: (with probability at least $1 - \frac{1}{n}$ every packet reaches its destination in $14n$ or fewer steps) for any particular π_i a fraction of at most $\frac{1}{n}$ of the algorithms A_j are bad - which of them are bad may differ from instance to instance.

For a permutation $\pi_i,\ 1 \leq i \leq n!$, let us call a deterministic oblivious routing algorithm A_j **good** if A_j routes π_i in $14d$ or fewer steps, and **bad** otherwise.

By our randomized routing result: (with probability at least $1 - \frac{1}{n}$ every packet reaches its destination in $14n$ or fewer steps) for any particular π_i a fraction of at most $\frac{1}{n}$ of the algorithms A_j are bad - which of them are bad may differ from instance to instance.

Experiment: Choose n^3 indices $i_1, \ldots, i_{n^3}$, randomly, independently and uniformly from the set $\{1, \ldots, n^n\}$. We show that the set of deterministic algorithms

$$\mathcal{A} = \{\mathcal{A}_{i_1}, \ldots, \mathcal{A}_{i_{n^3}}\}$$

is an efficient routing scheme with a positive probability. This will imply that an efficient routing scheme exists for any $n = 2^d$.

For a permutation $\pi_i,\ 1 \leq i \leq n!$, let us call a deterministic oblivious routing algorithm A_j **good** if A_j routes π_i in $14d$ or fewer steps, and **bad** otherwise.

By our randomized routing result: (with probability at least $1 - \frac{1}{n}$ every packet reaches its destination in $14n$ or fewer steps) for any particular π_i a fraction of at most $\frac{1}{n}$ of the algorithms A_j are bad - which of them are bad may differ from instance to instance.

Experiment: Choose n^3 indices $i_1, \ldots, i_{n^3}$, randomly, independently and uniformly from the set $\{1, \ldots, n^n\}$. We show that the set of deterministic algorithms

$$\mathcal{A} = \{\mathcal{A}_{i_1}, \ldots, \mathcal{A}_{i_{n^3}}\}$$

is an efficient routing scheme with a positive probability. This will imply that an efficient routing scheme exists for any $n = 2^d$.

For any π_i, a fraction of at most $\frac{1}{n}$ of the algorithms $\mathcal{A}_1, \ldots, \mathcal{A}_{n^n}$ is bad. Therefore, the expected number of algorithms in $\mathcal{A}$ that are bad for π_i is at most $n^3 \cdot \frac{1}{n} = n^2$.

Let the indicator variable X_j be 1 if A_{i_j} is bad, $1 \leq j \leq n^3$, and 0 otherwise.

Let the indicator variable X_j be 1 if A_{i_j} is bad, $1 \leq j \leq n^3$, and 0 otherwise. Thus $\mathbf{E}[\sum_j X_j] \leq n^2$.

Let the indicator variable X_j be 1 if A_{i_j} is bad, $1 \leq j \leq n^3$, and 0 otherwise. Thus $\mathbf{E}[\sum_j X_j] \leq n^2$. Since X_j are independent, we may apply Chernoff bound on $X = \sum A_i$ to get

$$\Pr[X \geq (1+\delta)\mu] \leq F^+(\mu, \delta) < e^{-\frac{\mu\delta^2}{4}}$$

(Chapter 6, page 10), to obtain (for $\mu = n^2, \delta = 1$) that the probability that more than $2n^2$ of the algorithms in $\mathcal{A}$ are bad for π_i is $\leq e^{\frac{-n^2}{4}}$. Let $\mathcal{B}_i$ denote the **bad** event that more than $2n^2$ algorithms in $\mathcal{A}$ are bad for π_i.

Let the indicator variable X_j be 1 if A_{i_j} is bad, $1 \leq j \leq n^3$, and 0 otherwise. Thus $\mathbf{E}[\sum_j X_j] \leq n^2$. Since X_j are independent, we may apply Chernoff bound on $X = \sum A_i$ to get

$$\Pr[X \geq (1+\delta)\mu] \leq F^+(\mu, \delta) < e^{-\frac{\mu\delta^2}{4}}$$

(Chapter 6, page 10), to obtain (for $\mu = n^2, \delta = 1$) that the probability that more than $2n^2$ of the algorithms in $\mathcal{A}$ are bad for π_i is $\leq e^{\frac{-n^2}{4}}$. Let $\mathcal{B}_i$ denote the **bad** event that more than $2n^2$ algorithms in $\mathcal{A}$ are bad for π_i. Then, for $n \geq 4$

$$Pr[\bigcup_{i=1}^{n!} \mathcal{B}_i] \leq \sum_{i=1}^{n!} Pr(\mathcal{B}_i) \leq n!\, e^{-\frac{n^2}{4}} < 1 \text{ (by Stirling's formula for } n!)$$

Let the indicator variable X_j be 1 if A_{i_j} is bad, $1 \leq j \leq n^3$, and 0 otherwise. Thus $\mathbf{E}[\sum_j X_j] \leq n^2$. Since X_j are independent, we may apply Chernoff bound on $X = \sum A_i$ to get

$$\Pr[X \geq (1+\delta)\mu] \leq F^+(\mu, \delta) < e^{-\frac{\mu\delta^2}{4}}$$

(Chapter 6, page 10), to obtain (for $\mu = n^2, \delta = 1$) that the probability that more than $2n^2$ of the algorithms in $\mathcal{A}$ are bad for π_i is $\leq e^{\frac{-n^2}{4}}$. Let $\mathcal{B}_i$ denote the **bad** event that more than $2n^2$ algorithms in $\mathcal{A}$ are bad for π_i. Then, for $n \geq 4$

$$Pr[\bigcup_{i=1}^{n!} \mathcal{B}_i] \leq \sum_{i=1}^{n!} Pr(\mathcal{B}_i) \leq n!\, e^{-\frac{n^2}{4}} < 1 \text{ (by Stirling's formula for } n!)$$

Therefore, with positive probability, not more than $2n^2$ algorithms in $\mathcal{A}$ are bad for any π_i.

Let the indicator variable X_j be 1 if A_{i_j} is bad, $1 \leq j \leq n^3$, and 0 otherwise. Thus $\mathbf{E}[\sum_j X_j] \leq n^2$. Since X_j are independent, we may apply Chernoff bound on $X = \sum A_i$ to get

$$\Pr[X \geq (1+\delta)\mu] \leq F^+(\mu, \delta) < e^{-\frac{\mu\delta^2}{4}}$$

(Chapter 6, page 10), to obtain (for $\mu = n^2, \delta = 1$) that the probability that more than $2n^2$ of the algorithms in $\mathcal{A}$ are bad for π_i is $\leq e^{\frac{-n^2}{4}}$. Let $\mathcal{B}_i$ denote the **bad** event that more than $2n^2$ algorithms in $\mathcal{A}$ are bad for π_i. Then, for $n \geq 4$

$$Pr[\bigcup_{i=1}^{n!} \mathcal{B}_i] \leq \sum_{i=1}^{n!} Pr(\mathcal{B}_i) \leq n!\, e^{-\frac{n^2}{4}} < 1 \text{ (by Stirling's formula for } n!)$$

Therefore, with positive probability, not more than $2n^2$ algorithms in $\mathcal{A}$ are bad for any π_i. Hence, there exists a subset of n^3 algorithms from $\{\mathcal{A}_1, \ldots, \mathcal{A}_{n^n}\}$ with the property that at most $2n^2$ in this subset are bad for any π_i.

Let the indicator variable X_j be 1 if A_{i_j} is bad, $1 \leq j \leq n^3$, and 0 otherwise. Thus $\mathbf{E}[\sum_j X_j] \leq n^2$. Since X_j are independent, we may apply Chernoff bound on $X = \sum A_i$ to get

$$\Pr[X \geq (1+\delta)\mu] \leq F^+(\mu, \delta) < e^{-\frac{\mu\delta^2}{4}}$$

(Chapter 6, page 10), to obtain (for $\mu = n^2, \delta = 1$) that the probability that more than $2n^2$ of the algorithms in $\mathcal{A}$ are bad for π_i is $\leq e^{\frac{-n^2}{4}}$. Let $\mathcal{B}_i$ denote the **bad** event that more than $2n^2$ algorithms in $\mathcal{A}$ are bad for π_i. Then, for $n \geq 4$

$$Pr[\bigcup_{i=1}^{n!} \mathcal{B}_i] \leq \sum_{i=1}^{n!} Pr(\mathcal{B}_i) \leq n!\, e^{-\frac{n^2}{4}} < 1 \text{ (by Stirling's formula for } n!)$$

Therefore, with positive probability, not more than $2n^2$ algorithms in $\mathcal{A}$ are bad for any π_i. Hence, there exists a subset of n^3 algorithms from $\{\mathcal{A}_1, \ldots, \mathcal{A}_{n^n}\}$ with the property that at most $2n^2$ in this subset are bad for any π_i.

Let us denote this subset $\mathcal{B} = \{B_{i_1}, \ldots, B_{i_{n^3}}\}$. $\mathcal{B}$ is an efficient routing scheme: for any π_i a randomly chosen algorithm form $\mathcal{B}$ fails to route π_i within $14n$ steps with probability at most $\frac{2n^2}{n^3} = \frac{2}{n}$.

THE LOVÁSZ LOCAL LEMMA

THE LOVÁSZ LOCAL LEMMA - MOTIVATION I

THE LOVÁSZ LOCAL LEMMA - MOTIVATION I

This is one of the most elegant and useful tools to apply the probabilistic method.

THE LOVÁSZ LOCAL LEMMA - MOTIVATION I

This is one of the most elegant and useful tools to apply the probabilistic method.
Suppose that we have a finite set A of bad events such that each of them may not happen with a non-zero probability. We want to show that under certain circumstances none of these events happen.

THE LOVÁSZ LOCAL LEMMA - MOTIVATION I

This is one of the most elegant and useful tools to apply the probabilistic method. Suppose that we have a finite set $\mathcal{A}$ of bad events such that each of them may not happen with a non-zero probability. We want to show that under certain circumstances none of these events happen.

This is easy to show for the case events are independent. Indeed, in such a case

$$Pr\left[\bigcap_{A\in\mathcal{A}} \bar{A}\right] = \prod_{A\in\mathcal{A}} Pr[\bar{A}] > 0.$$

THE LOVÁSZ LOCAL LEMMA - MOTIVATION I

This is one of the most elegant and useful tools to apply the probabilistic method. Suppose that we have a finite set $\mathcal{A}$ of bad events such that each of them may not happen with a non-zero probability. We want to show that under certain circumstances none of these events happen.

This is easy to show for the case events are independent. Indeed, in such a case

$$Pr\left[\bigcap_{A \in \mathcal{A}} \bar{A}\right] = \prod_{A \in \mathcal{A}} Pr[\bar{A}] > 0.$$

Lovász Local lemma handle the situation for the case where events are generally not independent of each other, but each collection of events that are not independent of some particular event A has low total probability.

THE LOVÁSZ LOCAL LEMMA - MOTIVATION I

This is one of the most elegant and useful tools to apply the probabilistic method. Suppose that we have a finite set $\mathcal{A}$ of bad events such that each of them may not happen with a non-zero probability. We want to show that under certain circumstances none of these events happen.

This is easy to show for the case events are independent. Indeed, in such a case

$$Pr\left[\bigcap_{A \in \mathcal{A}} \bar{A}\right] = \prod_{A \in \mathcal{A}} Pr[\bar{A}] > 0.$$

Lovász Local lemma handle the situation for the case where events are generally not independent of each other, but each collection of events that are not independent of some particular event A has low total probability.

The proof of the original version of Lovász Local Lemma was non-constructive - it gave no guidance how to find a particular outcome that makes all the events false.

THE LOVÁSZ LOCAL LEMMA - MOTIVATION I

This is one of the most elegant and useful tools to apply the probabilistic method. Suppose that we have a finite set $\mathcal{A}$ of bad events such that each of them may not happen with a non-zero probability. We want to show that under certain circumstances none of these events happen.

This is easy to show for the case events are independent. Indeed, in such a case

$$Pr\left[\bigcap_{A \in \mathcal{A}} \bar{A}\right] = \prod_{A \in \mathcal{A}} Pr[\bar{A}] > 0.$$

Lovász Local lemma handle the situation for the case where events are generally not independent of each other, but each collection of events that are not independent of some particular event A has low total probability.

The proof of the original version of Lovász Local Lemma was non-constructive - it gave no guidance how to find a particular outcome that makes all the events false.

Later, it has been shown that when events are determined by some underlying sets of independent variables and independence between two events is detected by having non-overlapping sets of underlying variables, an actual solution can be found in polynomial expected time.

THE LOWÁSZ LOCAL LEMMA - SYMMETRIC VERSION

THE LOWÁSZ LOCAL LEMMA - SYMMETRIC VERSION

In order to formulate so called symmetric version of the lemma we need the following definition of mutual independence. An event E is mutually independent of events $E_1, \ldots, E_n$, if for any $I \subseteq [1, n]$,

$$Pr\left(E \mid \bigcap_{i \in I} E_i\right) = Pr(E).$$

Dependency between events can be represented in terms of a dependency graph.

THE LOWÁSZ LOCAL LEMMA - SYMMETRIC VERSION

In order to formulate so called symmetric version of the lemma we need the following definition of mutual independence. An event E is mutually independent of events $E_1, \ldots, E_n$, if for any $I \subseteq [1, n]$,

$$Pr\left(E \mid \bigcap_{i \in I} E_i\right) = Pr(E).$$

Dependency between events can be represented in terms of a dependency graph.

Definition: Dependency graph for a set of events $E_1, \ldots, E_n$ is a graph $G = (V, E)$ such that $V = \{1, \ldots, n\}$ and for $i = 1, \ldots, n$, the event E_i is mutually independent of the events $\{E_j \mid (i, j) \notin E\}$.

THE LOWÁSZ LOCAL LEMMA - SYMMETRIC VERSION

In order to formulate so called symmetric version of the lemma we need the following definition of mutual independence. An event E is mutually independent of events $E_1, \dots, E_n$, if for any $I \subseteq [1, n]$,

$$Pr\left(E \mid \bigcap_{i \in I} E_i\right) = Pr(E).$$

Dependency between events can be represented in terms of a dependency graph.

Definition: Dependency graph for a set of events $E_1, \dots, E_n$ is a graph $G = (V, E)$ such that $V = \{1, \dots, n\}$ and for $i = 1, \dots, n$, the event E_i is mutually independent of the events $\{E_j \mid (i, j) \notin E\}$.

Lovász Local Lemma Let $E_1, \dots, E_n$ be events and p, d be fixed numbers such that

- for all i, $Pr(E_i) \leq p$;

THE LOWÁSZ LOCAL LEMMA - SYMMETRIC VERSION

In order to formulate so called symmetric version of the lemma we need the following definition of mutual independence. An event E is mutually independent of events $E_1, \ldots, E_n$, if for any $I \subseteq [1, n]$,

$$Pr\left(E \mid \bigcap_{i \in I} E_i\right) = Pr(E).$$

Dependency between events can be represented in terms of a dependency graph.

Definition: Dependency graph for a set of events $E_1, \ldots, E_n$ is a graph $G = (V, E)$ such that $V = \{1, \ldots, n\}$ and for $i = 1, \ldots, n$, the event E_i is mutually independent of the events $\{E_j \mid (i, j) \notin E\}$.

Lovász Local Lemma Let $E_1, \ldots, E_n$ be events and p, d be fixed numbers such that

- for all i, $Pr(E_i) \leq p$;
- the degree of the dependency graph given by $E_1, \ldots, E_n$ is bounded by a d;

THE LOWÁSZ LOCAL LEMMA - SYMMETRIC VERSION

In order to formulate so called symmetric version of the lemma we need the following definition of mutual independence. An event E is mutually independent of events $E_1, \ldots, E_n$, if for any $I \subseteq [1, n]$,

$$Pr\left(E \mid \bigcap_{i \in I} E_i\right) = Pr(E).$$

Dependency between events can be represented in terms of a dependency graph.

Definition: Dependency graph for a set of events $E_1, \ldots, E_n$ is a graph $G = (V, E)$ such that $V = \{1, \ldots, n\}$ and for $i = 1, \ldots, n$, the event E_i is mutually independent of the events $\{E_j \mid (i, j) \notin E\}$.

Lovász Local Lemma Let $E_1, \ldots, E_n$ be events and p, d be fixed numbers such that

- for all i, $Pr(E_i) \leq p$;
- the degree of the dependency graph given by $E_1, \ldots, E_n$ is bounded by a d;
- $4dp \leq 1$.

THE LOWÁSZ LOCAL LEMMA - SYMMETRIC VERSION

In order to formulate so called symmetric version of the lemma we need the following definition of mutual independence. An event E is mutually independent of events $E_1, \dots, E_n$, if for any $I \subseteq [1, n]$,

$$Pr\left(E \mid \bigcap_{i \in I} E_i\right) = Pr(E).$$

Dependency between events can be represented in terms of a dependency graph.

Definition: Dependency graph for a set of events $E_1, \dots, E_n$ is a graph $G = (V, E)$ such that $V = \{1, \dots, n\}$ and for $i = 1, \dots, n$, the event E_i is mutually independent of the events $\{E_j \mid (i, j) \notin E\}$.

Lovász Local Lemma Let $E_1, \dots, E_n$ be events and p, d be fixed numbers such that

- for all i, $Pr(E_i) \leq p$;
- the degree of the dependency graph given by $E_1, \dots, E_n$ is bounded by a d;
- $4dp \leq 1$.

Then $Pr\left(\bigcap_{i=1}^{n} \bar{E}_i\right) > 0$.

APPLICATION - EDGE-DISJOINT PATHS

APPLICATION - EDGE-DISJOINT PATHS

- Assume that n pairs of users need to communicate using edge-disjoint paths, from a fixed set of paths, on a given network .

APPLICATION - EDGE-DISJOINT PATHS

- Assume that n pairs of users need to communicate using edge-disjoint paths, from a fixed set of paths, on a given network .
- Assume that each ith pair of users can choose a path from a collection F_i of m paths for a fixed m.

APPLICATION - EDGE-DISJOINT PATHS

- Assume that n pairs of users need to communicate using edge-disjoint paths, from a fixed set of paths, on a given network .
- Assume that each ith pair of users can choose a path from a collection F_i of m paths for a fixed m.
- We show, using simple version of Lovász local lemma, that if possible paths do not share too many edges, then there is a way to choose n edge-disjoint paths connecting the n given pairs.

APPLICATION - EDGE-DISJOINT PATHS

- Assume that n pairs of users need to communicate using edge-disjoint paths, from a fixed set of paths, on a given network .
- Assume that each ith pair of users can choose a path from a collection F_i of m paths for a fixed m.
- We show, using simple version of Lovász local lemma, that if possible paths do not share too many edges, then there is a way to choose n edge-disjoint paths connecting the n given pairs.

Theorem: If, for a fixed k, any path in any F_i shares edges with no more than k paths in any $F_j, j \neq i$, and $8nk/m \leq 1$, then there is a way to choose n edge-disjoint paths connecting the n given pairs.

PROOF of THEOREM

PROOF of THEOREM

- Let each ith pair chooses a path from F_i randomly, with probability $\frac{1}{m}$.

PROOF of THEOREM

- Let each ith pair chooses a path from F_i randomly, with probability $\frac{1}{m}$.
- Let $E_{i,j}$ be event that paths chosen by pairs i and j share at least one edge.

PROOF of THEOREM

- Let each ith pair chooses a path from F_i randomly, with probability $\frac{1}{m}$.
- Let $E_{i,j}$ be event that paths chosen by pairs i and j share at least one edge.
- Since for any i and $j \neq i$ the path chosen from F_i shares edges with at most k paths in any $F_j, j \neq i$, we have

$$p = \Pr(E_{i,j}) \leq \frac{k}{m}.$$

PROOF of THEOREM

- Let each ith pair chooses a path from F_i randomly, with probability $\frac{1}{m}$.
- Let $E_{i,j}$ be event that paths chosen by pairs i and j share at least one edge.
- Since for any i and $j \neq i$ the path chosen from F_i shares edges with at most k paths in any $F_j, j \neq i$, we have

$$p = \Pr(E_{i,j}) \leq \frac{k}{m}.$$

- Let d be the degree of the dependency graph of all events $E_{i,j}$. Since the event $E_{i,j}$ is independent of all events $E_{i',j'}$, when $i' \notin \{i,j\}$ and $j' \notin \{i,j\}$, we have $d < 2n$.

PROOF of THEOREM

- Let each ith pair chooses a path from F_i randomly, with probability $\frac{1}{m}$.
- Let $E_{i,j}$ be event that paths chosen by pairs i and j share at least one edge.
- Since for any i and $j \neq i$ the path chosen from F_i shares edges with at most k paths in any $F_j, j \neq i$, we have

$$p = \Pr(E_{i,j}) \leq \frac{k}{m}.$$

- Let d be the degree of the dependency graph of all events $E_{i,j}$. Since the event $E_{i,j}$ is independent of all events $E_{i',j'}$, when $i' \notin \{i,j\}$ and $j' \notin \{i,j\}$, we have $d < 2n$.
- Since

$$4dp < \frac{8nk}{m} \leq 1$$

all conditions of the Lovász local lemma are satisfied and therefore

$$Pr\left(\bigcap_{i \neq j} \bar{E}_{i,j}\right) > 0$$

PROOF of THEOREM

- Let each ith pair chooses a path from F_i randomly, with probability $\frac{1}{m}$.
- Let $E_{i,j}$ be event that paths chosen by pairs i and j share at least one edge.
- Since for any i and $j \neq i$ the path chosen from F_i shares edges with at most k paths in any $F_j, j \neq i$, we have

$$p = \Pr(E_{i,j}) \leq \frac{k}{m}.$$

- Let d be the degree of the dependency graph of all events $E_{i,j}$. Since the event $E_{i,j}$ is independent of all events $E_{i',j'}$, when $i' \notin \{i,j\}$ and $j' \notin \{i,j\}$, we have $d < 2n$.
- Since

$$4dp < \frac{8nk}{m} \leq 1$$

all conditions of the Lovász local lemma are satisfied and therefore

$$Pr\left(\bigcap_{i \neq j} \bar{E}_{i,j}\right) > 0$$

and therefore there is such a choice of n paths that are edge-disjoint.

THE LOVÁSZ LOCAL LEMMA - GENERAL CASE

THE LOVÁSZ LOCAL LEMMA - GENERAL CASE

Suppose we have n events each of which occur with probability at most $\frac{1}{2}$. Let each of these events corresponds to one of n ways in which a probabilistic experiment could fail.

THE LOVÁSZ LOCAL LEMMA - GENERAL CASE

Suppose we have n events each of which occur with probability at most $\frac{1}{2}$. Let each of these events corresponds to one of n ways in which a probabilistic experiment could fail.

If the event were independent, we could then assert that at such an experiment with probability at least 2^{-n}, none of the events occurs.

THE LOVÁSZ LOCAL LEMMA - GENERAL CASE

Suppose we have n events each of which occur with probability at most $\frac{1}{2}$. Let each of these events corresponds to one of n ways in which a probabilistic experiment could fail.

If the event were independent, we could then assert that at such an experiment with probability at least 2^{-n}, none of the events occurs.

The Lovász Local Lemma generalizes the above setting and result to the case where each of the events is independent of all but small number of other events.

THE LOVÁSZ LOCAL LEMMA - GENERAL CASE

Suppose we have n events each of which occur with probability at most $\frac{1}{2}$. Let each of these events corresponds to one of n ways in which a probabilistic experiment could fail.

If the event were independent, we could then assert that at such an experiment with probability at least 2^{-n}, none of the events occurs.

The Lovász Local Lemma generalizes the above setting and result to the case where each of the events is independent of all but small number of other events.

Lovász Local Lemma: Let $G = (V, E)$ be the dependency graph for events $\xi_1, \ldots, \xi_n$ in a probability space.

THE LOVÁSZ LOCAL LEMMA - GENERAL CASE

Suppose we have n events each of which occur with probability at most $\frac{1}{2}$. Let each of these events corresponds to one of n ways in which a probabilistic experiment could fail.

If the event were independent, we could then assert that at such an experiment with probability at least 2^{-n}, none of the events occurs.

The Lovász Local Lemma generalizes the above setting and result to the case where each of the events is independent of all but small number of other events.

Lovász Local Lemma: Let $G = (V, E)$ be the dependency graph for events $\xi_1, \ldots, \xi_n$ in a probability space. Suppose that there exist $x_i \in [0, 1]$, for $1 \leq i \leq n$, such that

$$\Pr[\xi_i] \leq \prod_{(i,j) \in E} (1 - x_j).$$

THE LOVÁSZ LOCAL LEMMA - GENERAL CASE

Suppose we have n events each of which occur with probability at most $\frac{1}{2}$. Let each of these events corresponds to one of n ways in which a probabilistic experiment could fail.

If the event were independent, we could then assert that at such an experiment with probability at least 2^{-n}, none of the events occurs.

The Lovász Local Lemma generalizes the above setting and result to the case where each of the events is independent of all but small number of other events.

Lovász Local Lemma: Let $G = (V, E)$ be the dependency graph for events $\xi_1, \ldots, \xi_n$ in a probability space. Suppose that there exist $x_i \in [0, 1]$, for $1 \leq i \leq n$, such that

$$\Pr[\xi_i] \leq \prod_{(i,j) \in E} (1 - x_j).$$

Then

$$\Pr[\bigcap_{i=1}^{n} \bar{\xi}_i] \geq \prod_{i=1}^{n} (1 - x_i).$$

PAUL ERDÖS

PAUL ERDÖS

- Paul Erdös, a Hungarian mathematician, died in September 1996 at the age of 83.

PAUL ERDÖS

- Paul Erdös, a Hungarian mathematician, died in September 1996 at the age of 83.
- He was the most prolific mathematicians of the twentieth century with over 1500 papers written and more than 490 collaborators.

PAUL ERDÖS

- Paul Erdös, a Hungarian mathematician, died in September 1996 at the age of 83.
- He was the most prolific mathematicians of the twentieth century with over 1500 papers written and more than 490 collaborators.
- He can be seen as the founder (in 1947) of the probabilistic method.

THOUGHTS ABOUT ERDÖS

THOUGHTS ABOUT ERDÖS

- In our century, in which mathematics is so strongly dominating by "theory constructors" Erdös remained to be the "prince of problem solvers"

THOUGHTS ABOUT ERDÖS

- In our century, in which mathematics is so strongly dominating by "theory constructors" Erdös remained to be the "prince of problem solvers" and the "absolute monarch of problem posters/formulators". (E. Strauss)

THOUGHTS ABOUT ERDÖS

- In our century, in which mathematics is so strongly dominating by "theory constructors" Erdös remained to be the "prince of problem solvers" and the "absolute monarch of problem posters/formulators". (E. Strauss)
- Erdös had no job

THOUGHTS ABOUT ERDÖS

- In our century, in which mathematics is so strongly dominating by "theory constructors" Erdös remained to be the "prince of problem solvers" and the "absolute monarch of problem posters/formulators". (E. Strauss)
- Erdös had no job though he worked constantly.

THOUGHTS ABOUT ERDÖS

- In our century, in which mathematics is so strongly dominating by "theory constructors" Erdös remained to be the "prince of problem solvers" and the "absolute monarch of problem posters/formulators". (E. Strauss)

- Erdös had no job though he worked constantly. He had no home -

THOUGHTS ABOUT ERDÖS

- In our century, in which mathematics is so strongly dominating by "theory constructors" Erdös remained to be the "prince of problem solvers" and the "absolute monarch of problem posters/formulators". (E. Strauss)
- Erdös had no job though he worked constantly. He had no home -the world was his home.

THOUGHTS ABOUT ERDÖS

- In our century, in which mathematics is so strongly dominating by "theory constructors" Erdös remained to be the "prince of problem solvers" and the "absolute monarch of problem posters/formulators". (E. Strauss)

- Erdös had no job though he worked constantly. He had no home -the world was his home. Possessions were for him a nuisance,

THOUGHTS ABOUT ERDÖS

- In our century, in which mathematics is so strongly dominating by "theory constructors" Erdös remained to be the "prince of problem solvers" and the "absolute monarch of problem posters/formulators". (E. Strauss)

- Erdös had no job though he worked constantly. He had no home -the world was his home. Possessions were for him a nuisance, money a bore.

THOUGHTS ABOUT ERDÖS

- In our century, in which mathematics is so strongly dominating by "theory constructors" Erdös remained to be the "prince of problem solvers" and the "absolute monarch of problem posters/formulators". (E. Strauss)

- Erdös had no job though he worked constantly. He had no home -the world was his home. Possessions were for him a nuisance, money a bore.He lived "on a web of trust", traveling from Center to Center,

THOUGHTS ABOUT ERDÖS

- In our century, in which mathematics is so strongly dominating by "theory constructors" Erdös remained to be the "prince of problem solvers" and the "absolute monarch of problem posters/formulators". (E. Strauss)

- Erdös had no job though he worked constantly. He had no home -the world was his home. Possessions were for him a nuisance, money a bore.He lived "on a web of trust", traveling from Center to Center, spreading his mathematical pollen (pel in Czech). His enormous talents and energies were given entirely to the "Temple of Mathematics".

STORIES about ERDÖS

STORIES about ERDÖS

- It is six in the morning. The house is sleeping. I prove and conjecture.

STORIES about ERDÖS

- It is six in the morning. The house is sleeping. I prove and conjecture.

Paul Erdös in a letter to Vera Sós

STORY about the BOOK

STORY about the BOOK

- Story about The Book.

STORY about the BOOK

- Story about The Book. Erdös liked to talk about The Book, that contains all theorems of mathematics and for each of them one proof - beautiful, aesthetic and insightful - so called The Book proof.

STORY about the BOOK

- Story about The Book. Erdös liked to talk about The Book, that contains all theorems of mathematics and for each of them one proof - beautiful, aesthetic and insightful - so called The Book proof.
  Each time when some of his conjectures was resolved in "an ugly way" Erdös congratulated the prover, but added "let us now look for a Book proof".

STORY about the BOOK

- Story about The Book. Erdös liked to talk about The Book, that contains all theorems of mathematics and for each of them one proof - beautiful, aesthetic and insightful - so called The Book proof.
  Each time when some of his conjectures was resolved in "an ugly way" Erdös congratulated the prover, but added "let us now look for a Book proof".
- In 1985 he started his lecture in a math camp by saying: "You do not have to believe in the God, but you should believe in The Book".