Approximate Counting

Martin Dyer
School of Computing
University of Leeds

Berlin
June 6, 2005
Outline

1. Exact Counting
2. Approximate Counting
3. Direct Monte Carlo
   - Coupling
   - Eigenvalue estimates and relatives
4. Markov Chain Monte Carlo
5. Hardness of approximate counting
1. Exact Counting

2. Approximate Counting

3. Direct Monte Carlo

4. Markov Chain Monte Carlo
   - Coupling
   - Eigenvalue estimates and relatives

5. Hardness of approximate counting
Outline

1. Exact Counting
2. Approximate Counting
3. Direct Monte Carlo
4. Markov Chain Monte Carlo
   • Coupling
   • Eigenvalue estimates and relatives
5. Hardness of approximate counting
Outline

1. Exact Counting
2. Approximate Counting
3. Direct Monte Carlo
4. Markov Chain Monte Carlo
   - Coupling
     - Eigenvalue estimates and relatives
5. Hardness of approximate counting
1. Exact Counting
2. Approximate Counting
3. Direct Monte Carlo
4. Markov Chain Monte Carlo
   - Coupling
   - Eigenvalue estimates and relatives
5. Hardness of approximate counting
Outline

1. Exact Counting
2. Approximate Counting
3. Direct Monte Carlo
   - Coupling
   - Eigenvalue estimates and relatives
4. Markov Chain Monte Carlo
5. Hardness of approximate counting
Counting Problems

Counting combinatorially-defined sets is well studied.

For example:

- **Chromatic polynomial**: Count the number of $q$-colourings of a graph $G$.
- **Zero-one permanent**: Count the number of perfect matchings of a bipartite graph $G$.

Sometimes the objects are weighted in some way, and we wish to determine the sum of the weights over all objects (partition function).

The general permanent is an example. Here $G$ is a non-negative edge weighted graph. The weight of each matching is the product of the weights of all its edges.
Counting combinatorially-defined sets is well studied. 

For example:

- **Chromatic polynomial**: Count the number of $q$-colourings of a graph $G$.
- **Zero-one permanent**: Count the number of perfect matchings of a bipartite graph $G$.

Sometimes the objects are weighted in some way, and we wish to determine the sum of the weights over all objects (**partition function**).

The **general permanent** is an example. Here $G$ is a non-negative edge weighted graph. The weight of each matching is the product of the weights of all its edges.
Counting Problems

Counting combinatorially-defined sets is well studied.

For example:

- **Chromatic polynomial**: Count the number of $q$-colourings of a graph $G$.

- **Zero-one permanent**: Count the number of perfect matchings of a bipartite graph $G$.

Sometimes the objects are weighted in some way, and we wish to determine the sum of the weights over all objects (partition function).

The general permanent is an example. Here $G$ is a non-negative edge weighted graph. The weight of each matching is the product of the weights of all its edges.
Counting combinatorially-defined sets is well studied.

For example:

- **Chromatic polynomial**: Count the number of \( q \)-colourings of a graph \( G \).

- **Zero-one permanent**: Count the number of perfect matchings of a bipartite graph \( G \).

Sometimes the objects are weighted in some way, and we wish to determine the sum of the weights over all objects (\textit{partition function}).

The general permanent is an example. Here \( G \) is a non-negative edge weighted graph. The weight of each matching is the product of the weights of all its edges.
Counting Problems

Counting combinatorially-defined sets is well studied.

For example:

- **Chromatic polynomial**: Count the number of \( q \)-colourings of a graph \( G \).

- **Zero-one permanent**: Count the number of perfect matchings of a bipartite graph \( G \).

Sometimes the objects are weighted in some way, and we wish to determine the sum of the weights over all objects (partition function).

The **general permanent** is an example. Here \( G \) is a non-negative edge weighted graph. The weight of each matching is the product of the weights of all its edges.
Why study counting problems?

There are many mathematical reasons, but a few practical reasons are

- Counting problems are a clean formulation of the notion of integration, which is part of many mathematical models of the real world. An example from the optimization world is Stochastic Programming.
- Counting is intimately related to sampling, which is used extensively in Statistics and elsewhere. An example of a problem from this domain is sampling contingency tables.
- Discrete counting and sampling problems arise naturally in models from statistical physics. Examples are the Ising and Potts models.
Why study counting problems?

There are many mathematical reasons, but a few practical reasons are:

- Counting problems are a clean formulation of the notion of integration, which is part of many mathematical models of the real world. An example from the optimization world is Stochastic Programming.
- Counting is intimately related to sampling, which is used extensively in Statistics and elsewhere. An example of a problem from this domain is sampling contingency tables.
- Discrete counting and sampling problems arise naturally in models from statistical physics. Examples are the Ising and Potts models.
Why study counting problems?

There are many mathematical reasons, but a few practical reasons are:

- Counting problems are a clean formulation of the notion of integration, which is part of many mathematical models of the real world. An example from the optimization world is Stochastic Programming.

- Counting is intimately related to sampling, which is used extensively in Statistics and elsewhere. An example of a problem from this domain is sampling contingency tables.

- Discrete counting and sampling problems arise naturally in models from statistical physics. Examples are the Ising and Potts models.
Why study counting problems?

There are many mathematical reasons, but a few practical reasons are:

- Counting problems are a clean formulation of the notion of integration, which is part of many mathematical models of the real world. An example from the optimization world is Stochastic Programming.
- Counting is intimately related to sampling, which is used extensively in Statistics and elsewhere. An example of a problem from this domain is sampling contingency tables.
- Discrete counting and sampling problems arise naturally in models from statistical physics. Examples are the Ising and Potts models.
A problem is a predicate $\Phi(x, y)$, where $x, y$ are strings over an alphabet $\Sigma$, such that $\Phi(x, y)$ implies $|y| = O(|x|^k)$ for some constant $k \geq 0$.

- We require $\Phi$ to be polynomial time, i.e. $\Phi(x, y)$ can be decided in time polynomial in $|x|$ for all pairs $x, y$.
- Here $x$ encodes the instance (e.g. a graph $G$) and $y$ encodes a witness (e.g. a perfect matching in $G$).

Such $\Phi$ determine the class $\text{NP}$. If the predicate $\exists y \, \Phi(x, y)$ is computable in time polynomial in $|x|$, then $\Phi \in \text{P}$.

Determining $|\{y : \Phi(x, y)\}|$ for given $x$ determines the class $\text{\#P}$. This class was introduced by Valiant, 1979.
A problem is a predicate $\Phi(x, y)$, where $x, y$ are strings over an alphabet $\Sigma$, such that $\Phi(x, y)$ implies $|y| = O(|x|^k)$ for some constant $k \geq 0$.

- We require $\Phi$ to be polynomial time, i.e. $\Phi(x, y)$ can be decided in time polynomial in $|x|$ for all pairs $x, y$.
- Here $x$ encodes the instance (e.g. a graph $G$) and $y$ encodes a witness (e.g. a perfect matching in $G$).

Such $\Phi$ determine the class $\text{NP}$. If the predicate $\exists y \, \Phi(x, y)$ is computable in time polynomial in $|x|$, then $\Phi \in \text{P}$.

Determining $|\{y : \Phi(x, y)\}|$ for given $x$ determines the class $\#\text{P}$. This class was introduced by Valiant, 1979.
The Class $\#P$

A problem is a predicate $\Phi(x, y)$, where $x, y$ are strings over an alphabet $\Sigma$, such that $\Phi(x, y)$ implies $|y| = O(|x|^k)$ for some constant $k \geq 0$.

- We require $\Phi$ to be polynomial time, i.e. $\Phi(x, y)$ can be decided in time polynomial in $|x|$ for all pairs $x, y$.

- Here $x$ encodes the instance (e.g. a graph $G$) and $y$ encodes a witness (e.g. a perfect matching in $G$).

Such $\Phi$ determine the class $NP$. If the predicate $\exists y \, \Phi(x, y)$ is computable in time polynomial in $|x|$, then $\Phi \in P$.

Determining $|\{y : \Phi(x, y)\}|$ for given $x$ determines the class $\#P$. This class was introduced by Valiant, 1979.
The Class $\text{\#P}$

A problem is a predicate $\Phi(x, y)$, where $x, y$ are strings over an alphabet $\Sigma$, such that $\Phi(x, y)$ implies $|y| = O(|x|^k)$ for some constant $k \geq 0$.

- We require $\Phi$ to be polynomial time, i.e. $\Phi(x, y)$ can be decided in time polynomial in $|x|$ for all pairs $x, y$.
- Here $x$ encodes the instance (e.g. a graph $G$) and $y$ encodes a witness (e.g. a perfect matching in $G$).

Such $\Phi$ determine the class $\text{NP}$. If the predicate $\exists y \Phi(x, y)$ is computable in time polynomial in $|x|$, then $\Phi \in \text{P}$.

Determining $|\{ y : \Phi(x, y) \}|$ for given $x$ determines the class $\text{\#P}$. This class was introduced by Valiant, 1979.
A problem is a predicate $\Phi(x, y)$, where $x, y$ are strings over an alphabet $\Sigma$, such that $\Phi(x, y)$ implies $|y| = O(|x|^k)$ for some constant $k \geq 0$.

- We require $\Phi$ to be polynomial time, i.e. $\Phi(x, y)$ can be decided in time polynomial in $|x|$ for all pairs $x, y$.
- Here $x$ encodes the instance (e.g. a graph $G$) and $y$ encodes a witness (e.g. a perfect matching in $G$).

Such $\Phi$ determine the class $\text{NP}$. If the predicate $\exists y \, \Phi(x, y)$ is computable in time polynomial in $|x|$, then $\Phi \in \text{P}$.

Determining $|\{y : \Phi(x, y)\}|$ for given $x$ determines the class $\#\text{P}$. This class was introduced by Valiant, 1979.
Very few nontrivial exact counting problems are in $\mathsf{P}$. Examples are counting spanning trees in graphs and perfect matchings in planar graphs. (Valiant, 2004, generalises the latter.) Both rest on reducing the problem to computing the determinant of a matrix. There could perhaps be a deep connection.

Most combinatorial counting problems are $\#\mathsf{P}$-complete. For example, the 0-1 permanent (Valiant, 1979) is $\#\mathsf{P}$-complete.

Completeness in $\#\mathsf{P}$ is defined by Turing reduction. This means that a reduction from problem $A$ to problem $B$ can call an oracle for $A$ any polynomial number of times and perform any other polynomial time computation.
Very few nontrivial exact counting problems are in P. Examples are counting spanning trees in graphs and perfect matchings in planar graphs. (Valiant, 2004, generalises the latter.) Both rest on reducing the problem to computing the determinant of a matrix. There could perhaps be a deep connection.

Most combinatorial counting problems are \textbf{\#P-complete}. For example, the 0-1 permanent (Valiant, 1979) is \textbf{\#P-complete}.

Completeness in \textbf{\#P} is defined by Turing reduction. This means that a reduction from problem A to problem B can call an oracle for A any polynomial number of times and perform any other polynomial time computation.
Very few nontrivial exact counting problems are in P. Examples are counting spanning trees in graphs and perfect matchings in planar graphs. (Valiant, 2004, generalises the latter.) Both rest on reducing the problem to computing the determinant of a matrix. There could perhaps be a deep connection.

Most combinatorial counting problems are \#P-complete. For example, the 0-1 permanent (Valiant, 1979) is \#P-complete.

Completeness in \#P is defined by Turing reduction. This means that a reduction from problem A to problem B can call an oracle for A any polynomial number of times and perform any other polynomial time computation.
Proving \#P-Completeness

Just as the generic \textbf{NP}-complete problem is \textbf{SAT}, the generic \#P-complete problem is \#\textbf{SAT}.

\textbf{NP}-completeness proofs use \textit{gadgets} to reduce SAT, or some other \textbf{NP}-complete problem, to the problem under consideration.

Direct reductions from \#\textbf{SAT}, or some other \#P-complete problem, must be \textit{parsimonious}, i.e. preserve numbers of solutions. This often makes the proof more difficult than \textbf{NP}-completeness proofs.

However, since the reductions are Turing, there is an additional tool which makes up for this: \textit{interpolation}.
Proving \#P-Completeness

Just as the generic \textbf{NP}-complete problem is \textbf{SAT}, the generic \textbf{#P}-complete problem is \textbf{#SAT}.

\textbf{NP}-completeness proofs use \textit{gadgets} to reduce \textbf{SAT}, or some other \textbf{NP}-complete problem, to the problem under consideration.

Direct reductions from \textbf{#SAT}, or some other \textbf{#P}-complete problem, must be \textit{parsimonious}, i.e. preserve numbers of solutions. This often makes the proof more difficult than \textbf{NP}-completeness proofs.

However, since the reductions are Turing, there is an additional tool which makes up for this: \textit{interpolation}. 
Proving \#P-Completeness

Just as the generic \textbf{NP}-complete problem is \textbf{SAT}, the generic \#P-complete problem is \#\textbf{SAT}.

\textbf{NP}-completeness proofs use \textit{gadgets} to reduce \textbf{SAT}, or some other \textbf{NP}-complete problem, to the problem under consideration.

Direct reductions from \#\textbf{SAT}, or some other \#P-complete problem, must be \textit{parsimonious}, i.e. preserve numbers of solutions. This often makes the proof more difficult than \textbf{NP}-completeness proofs.

However, since the reductions are Turing, there is an additional tool which makes up for this: \textit{interpolation}. 
Proving \#P-Completeness

Just as the generic \textbf{NP}\-complete problem is \textbf{SAT}, the generic \textbf{\#P}\-complete problem is \textbf{\#SAT}.

\textbf{NP}\-completeness proofs use \textit{gadgets} to reduce \textbf{SAT}, or some other \textbf{NP}\-complete problem, to the problem under consideration.

Direct reductions from \textbf{\#SAT}, or some other \textbf{\#P}\-complete problem, must be \textit{parsimonious}, i.e. preserve numbers of solutions. This often makes the proof more difficult than \textbf{NP}\-completeness proofs.

However, since the reductions are Turing, there is an additional tool which makes up for this: \textit{interpolation}. 

Interpolation

Many combinatorial counting numbers are expressible as polynomials of some kind over some parameter of the instances, which we can vary in some fashion.

If some coefficient, or value, of this polynomial is known to be \#P-complete, we can recover it by solving a set of linear equations.

Example

Colouring a graph $G$ with $q$ colours is easy if $q > \Delta$, the maximum degree. This is not true for counting colourings. Let $q_0$ be any number of colours. Suppose counting is easy for $q > q_0$. Now we can evaluate the chromatic polynomial for any $n$ values of $q > q_0$, recover its coefficients, and hence the number of 3-colourings. Counting 3-colourings can be shown \#P-complete by parsimonious reduction, so we have a contradiction.
Interpolation

Many combinatorial counting numbers are expressible as polynomials of some kind over some parameter of the instances, which we can vary in some fashion.

If some coefficient, or value, of this polynomial is known to be \( \#P \)-complete, we can recover it by solving a set of linear equations.

**Example**

Colouring a graph \( G \) with \( q \) colours is easy if \( q > \Delta \), the maximum degree. This is not true for counting colourings. Let \( q_0 \) be any number of colours. Suppose counting is easy for \( q > q_0 \). Now we can evaluate the chromatic polynomial for any \( n \) values of \( q > q_0 \), recover its coefficients, and hence the number of 3-colourings. Counting 3-colourings can be shown \( \#P \)-complete by parsimonious reduction, so we have a contradiction.
Interpolation

Many combinatorial counting numbers are expressible as polynomials of some kind over some parameter of the instances, which we can vary in some fashion.

If some coefficient, or value, of this polynomial is known to be \#P-complete, we can recover it by solving a set of linear equations.

Example

Colouring a graph $G$ with $q$ colours is easy if $q > \Delta$, the maximum degree. This is not true for counting colourings. Let $q_0$ be any number of colours. Suppose counting is easy for $q > q_0$. Now we can evaluate the chromatic polynomial for any $n$ values of $q > q_0$, recover its coefficients, and hence the number of 3-colourings. Counting 3-colourings can be shown \#P-complete by parsimonious reduction, so we have a contradiction.
It is known that \(#P\)-completeness is a much stronger property than \(NP\)-completeness. Toda, 1992, showed that any problem in the polynomial time hierarchy can be solved using an oracle for solving any \(#P\)-complete problem.

It is tempting to think that the counting version of an \(NP\)-complete problem must obviously be \(#P\)-complete. This may be true, but is not known to be true. The difficulty lies in the different way in which the classes \(NP\) and \(#P\) are defined. This remains an open problem.

Thus it is possible to show that counting the number of Hamilton cycles in a graph is \(#P\)-complete, but unfortunately this does not follow directly from the fact that finding only one is \(NP\)-complete.
It is known that $\#P$-completeness is a much stronger property than $NP$-completeness. Toda, 1992, showed that any problem in the polynomial time hierarchy can be solved using an oracle for solving any $\#P$-complete problem.

It is tempting to think that the counting version of an $NP$-complete problem must obviously be $\#P$-complete. This may be true, but is not known to be true. The difficulty lies in the different way in which the classes $NP$ and $\#P$ are defined. This remains an open problem.

Thus it is possible to show that counting the number of Hamilton cycles in a graph is $\#P$-complete, but unfortunately this does not follow directly from the fact that finding only one is $NP$-complete.
It is known that \#P-completeness is a much stronger property than NP-completeness. Toda, 1992, showed that any problem in the polynomial time hierarchy can be solved using an oracle for solving any \#P-complete problem.

It is tempting to think that the counting version of an NP-complete problem must obviously be \#P-complete. This may be true, but is not known to be true. The difficulty lies in the different way in which the classes NP and \#P are defined. This remains an open problem.

Thus it is possible to show that counting the number of Hamilton cycles in a graph is \#P-complete, but unfortunately this does not follow directly from the fact that finding only one is NP-complete.
Approximate counting

Given the difficulty of exact counting, we naturally ask how well can we approximate a function in \( \#P \) ?

The appropriate notion of approximation, as for optimization problems, is relative approximation.

Why?

To see this, suppose for example we could estimate the number \( N(G) \) of \( q \)-colourings of an \( n \)-vertex graph \( G \) within a fixed additive integer error \( b(n) < 2^{n/2} \) for any \( q \geq 2 \). Let \( G' \) be \( G \) plus a disjoint independent set of size \( n \), so \( N(G') = q^n N(G) \). Estimate, by the given algorithm, \( \hat{N}(G') = N(G') \pm b(2n) = q^n N(G) \pm b(2n) \), so we may take \( \hat{N}(G) = [\hat{N}(G')/q^n] = [N(G) \pm b(2n)/q^n] = N(G) \), since \( b(2n) < 2^n \leq q^n \), where \([\cdot]\) denotes the nearest integer.
Approximate counting

Given the difficulty of exact counting, we naturally ask how well can we approximate a function in \( \text{#P} \)?

The appropriate notion of approximation, as for optimization problems, is relative approximation.

**Why?**

To see this, suppose for example we could estimate the number \( N(G) \) of \( q \)-colourings of an \( n \)-vertex graph \( G \) within a fixed additive integer error \( b(n) < 2^{n/2} \) for any \( q \geq 2 \). Let \( G' \) be \( G \) plus a disjoint independent set of size \( n \), so \( N(G') = q^n N(G) \). Estimate, by the given algorithm, \( \hat{N}(G') = N(G') \pm b(2n) = q^n N(G) \pm b(2n) \), so we may take \( \hat{N}(G) = \left[ \frac{\hat{N}(G')}{q^n} \right] = \left[ \frac{N(G) \pm b(2n)}{q^n} \right] = N(G) \), since \( b(2n) < 2^n \leq q^n \), where \( \left[ \cdot \right] \) denotes the nearest integer.
Approximate counting

Given the difficulty of exact counting, we naturally ask how well can we approximate a function in \( \#P \)?

The appropriate notion of approximation, as for optimization problems, is relative approximation.

Why?
To see this, suppose for example we could estimate the number \( N(G) \) of \( q \)-colourings of an \( n \)-vertex graph \( G \) within a fixed additive integer error \( b(n) < 2^{n/2} \) for any \( q \geq 2 \). Let \( G' \) be \( G \) plus a disjoint independent set of size \( n \), so \( N(G') = q^n N(G) \). Estimate, by the given algorithm, \( \hat{N}(G') = N(G') \pm b(2n) = q^n N(G) \pm b(2n) \), so we may take \( \hat{N}(G) = \left[ \hat{N}(G')/q^n \right] = \left[ N(G) \pm b(2n)/q^n \right] = N(G) \), since \( b(2n) < 2^n \leq q^n \), where \([·]\) denotes the nearest integer.
Deterministic approximation algorithms are rare for counting problems. Natural combinatorial problems do not seem to possess deterministic approximations computable in polynomial time.

Stockmeyer, 1985, showed that deterministic approximation is possible using an oracle from the second level of the polynomial time hierarchy. Thus it may be harder than NP.

On the other hand, Valiant and Vazirani, 1986, gave a simple argument which shows that randomized approximation is possible for any problem in $\mathbf{#P}$ given only an oracle for an NP-complete problem.
Deterministic approximation algorithms are rare for counting problems. Natural combinatorial problems do not seem to possess deterministic approximations computable in polynomial time.

Stockmeyer, 1985, showed that deterministic approximation is possible using an oracle from the second level of the polynomial time hierarchy. Thus it may be harder than $\text{NP}$.

On the other hand, Valiant and Vazirani, 1986, gave a simple argument which shows that randomized approximation is possible for any problem in $\#P$ given only an oracle for an $\text{NP}$-complete problem.
Deterministic approximation algorithms are rare for counting problems. Natural combinatorial problems do not seem to possess deterministic approximations computable in polynomial time.

Stockmeyer, 1985, showed that deterministic approximation is possible using an oracle from the second level of the polynomial time hierarchy. Thus it may be harder than $\text{NP}$.

On the other hand, Valiant and Vazirani, 1986, gave a simple argument which shows that randomized approximation is possible for any problem in $\text{#P}$ given only an oracle for an $\text{NP}$-complete problem.
Proof sketch

Any problem in \#P can be parsimoniously reduced to \#SAT, using the construction of Cook’s Theorem, so we need only consider \#SAT.

Represent an instance of \#SAT as a set of equations in $n$ Boolean variables. Choose a linear equation over $\mathbb{Z}_2$ in these variables uniformly at random. About half the \#SAT solutions will also satisfy the added equation. Since this augmented problem is in NP, we can test whether it is satisfiable using our NP oracle.

We keep bisecting the set of solutions in this way, by adding further equations, until the augmented problem becomes unsatisfiable. It is easy to see that this happens when the previous problem had about one solution. If we bisected $k$ times, then our estimate of the size of the original set is $2^k$. 
Proof sketch

Any problem in $\#P$ can be parsimoniously reduced to $\#SAT$, using the construction of Cook’s Theorem, so we need only consider $\#SAT$.

Represent an instance of $\#SAT$ as a set of equations in $n$ Boolean variables. Choose a linear equation over $\mathbb{Z}_2$ in these variables uniformly at random. About half the $\#SAT$ solutions will also satisfy the added equation. Since this augmented problem is in $NP$, we can test whether it is satisfiable using our $NP$ oracle.

We keep bisecting the set of solutions in this way, by adding further equations, until the augmented problem becomes unsatisfiable. It is easy to see that this happens when the previous problem had about one solution. If we bisected $k$ times, then our estimate of the size of the original set is $2^k$. 
Any problem in $\mathbf{\#P}$ can be parsimoniously reduced to $\mathbf{\#SAT}$, using the construction of Cook’s Theorem, so we need only consider $\mathbf{\#SAT}$.

Represent an instance of $\mathbf{\#SAT}$ as a set of equations in $n$ Boolean variables. Choose a linear equation over $\mathbb{Z}_2$ in these variables uniformly at random. About half the $\mathbf{\#SAT}$ solutions will also satisfy the added equation. Since this augmented problem is in $\mathbf{NP}$, we can test whether it is satisfiable using our $\mathbf{NP}$ oracle.

We keep bisecting the set of solutions in this way, by adding further equations, until the augmented problem becomes unsatisfiable. It is easy to see that this happens when the previous problem had about one solution. If we bisected $k$ times, then our estimate of the size of the original set is $2^k$. 
The fpras

We are led to the following definition of Karp and Luby, 1985. Let $N$ be the exact number of witnesses for the instance $x$ of the problem $\Phi$. A **fully polynomial randomized approximation scheme (fpras)** for $\Phi$ is a randomized algorithm which produces an estimate $\hat{N}$ such that

$$\Pr((1 - \varepsilon)N \leq \hat{N} \leq (1 + \varepsilon)N) \geq \frac{3}{4},$$

and runs in time polynomial in the size $|x|$ of the instance and $1/\varepsilon$.

- The polynomial dependence on $1/\varepsilon$, rather than $\log(1/\varepsilon)$ is again because otherwise we could compute exactly in polynomial time.
- The $\frac{3}{4}$ is an arbitrary probability greater than $\frac{1}{2}$. It can be made as close to 1 as desired at modest cost by taking repeated estimates and using the sample median.
The fpras

We are led to the following definition of Karp and Luby, 1985. Let \( N \) be the exact number of witnesses for the instance \( x \) of the problem \( \Phi \). A fully polynomial randomized approximation scheme (fpras) for \( \Phi \) is a randomized algorithm which produces an estimate \( \hat{N} \) such that

\[
\Pr\left((1 - \varepsilon)N \leq \hat{N} \leq (1 + \varepsilon)N\right) \geq \frac{3}{4},
\]

and runs in time polynomial in the size \( |x| \) of the instance and \( 1/\varepsilon \).

- The polynomial dependence on \( 1/\varepsilon \), rather than \( \log(1/\varepsilon) \) is again because otherwise we could compute exactly in polynomial time.

- The \( \frac{3}{4} \) is an arbitrary probability greater than \( \frac{1}{2} \). It can be made as close to 1 as desired at modest cost by taking repeated estimates and using the sample median.
We are led to the following definition of Karp and Luby, 1985. Let $N$ be the exact number of witnesses for the instance $x$ of the problem $\Phi$. A fully polynomial randomized approximation scheme (fpras) for $\Phi$ is a randomized algorithm which produces an estimate $\hat{N}$ such that

$$\Pr((1 - \varepsilon)N \leq \hat{N} \leq (1 + \varepsilon)N) \geq \frac{3}{4},$$

and runs in time polynomial in the size $|x|$ of the instance and $1/\varepsilon$.

- The polynomial dependence on $1/\varepsilon$, rather than $\log(1/\varepsilon)$ is again because otherwise we could compute exactly in polynomial time.
- The $\frac{3}{4}$ is an arbitrary probability greater than $\frac{1}{2}$. It can be made as close to $1$ as desired at modest cost by taking repeated estimates and using the sample median.
Jerrum, Valiant and Vazirani, 1986, showed existence of an fpras is equivalent to almost uniform sampling from the witness set in many cases. When their theorem does not apply, equivalence can often be shown, but is probably not true for all problems in $\#P$ (Dyer, Goldberg, Greenhill and Jerrum, 2000).

Strictly, the JVV equivalence holds only for self reducible problems. Essentially this means that there is always some partition of the witness set which gives smaller instances of the same problem.

Example: Independent sets in a graph

Choose any vertex $v$ of $G$. For all independent sets $I$, If $v \in I$, $I$ is in the graph given by deleting $v$ and its neighbours. If $v \notin I$, $I$ is in the graph given by deleting $v$ only.
Jerrum, Valiant and Vazirani, 1986, showed existence of an fpras is equivalent to almost uniform sampling from the witness set in many cases. When their theorem does not apply, equivalence can often be shown, but is probably not true for all problems in $\#P$ (Dyer, Goldberg, Greenhill and Jerrum, 2000).

Strictly, the JVV equivalence holds only for self reducible problems. Essentially this means that there is always some partition of the witness set which gives smaller instances of the same problem.

Example: Independent sets in a graph

Choose any vertex $v$ of $G$. For all independent sets $I$, if $v \in I$, $I$ is in the graph given by deleting $v$ and its neighbours. If $v \notin I$, $I$ is in the graph given by deleting $v$ only.
Equivalence to sampling

Jerrum, Valiant and Vazirani, 1986, showed existence of an fpras is equivalent to almost uniform sampling from the witness set in many cases. When their theorem does not apply, equivalence can often be shown, but is probably not true for all problems in $\#P$ (Dyer, Goldberg, Greenhill and Jerrum, 2000).

 Strictly, the JVV equivalence holds only for self reducible problems. Essentially this means that there is always some partition of the witness set which gives smaller instances of the same problem.

Example: Independent sets in a graph

Choose any vertex $v$ of $G$. For all independent sets $I$, If $v \in I$, $I$ is in the graph given by deleting $v$ and its neighbours. If $v \notin I$, $I$ is in the graph given by deleting $v$ only.
Techniques for sampling

There are two main approaches:

- Direct Monte Carlo (DMC)
  Also called “dart throwing” by Karp and Luby, 1983. The original technique, but not the most successful.
  Revived by D, 2003, for the knapsack problem and relatives.

- Markov chain Monte Carlo (MCMC)
  Generate using a random walk on the witness set having polynomial time convergence to an almost uniform random solution (rapid mixing). First proposed by Broder, 1986, to count perfect matchings in dense bipartite graphs. Broder’s algorithm was proved correct by Jerrum and Sinclair, 1988.
  Notable successes include volume computation, permanent.
Techniques for sampling

There are two main approaches:

- **Direct Monte Carlo (DMC)**
  Also called “dart throwing” by Karp and Luby, 1983. The original technique, but not the most successful. Revived by D, 2003, for the *knapsack* problem and relatives.

- **Markov chain Monte Carlo (MCMC)**
  Generate using a random walk on the witness set having polynomial time convergence to an almost uniform random solution (*rapid mixing*). First proposed by Broder, 1986, to count perfect matchings in dense bipartite graphs. Broder’s algorithm was proved correct by Jerrum and Sinclair, 1988. Notable successes include volume computation, permanent.
Techniques for sampling

There are two main approaches:

- **Direct Monte Carlo (DMC)**
  Also called “dart throwing” by Karp and Luby, 1983. The original technique, but not the most successful. Revived by D, 2003, for the *knapsack* problem and relatives.

- **Markov chain Monte Carlo (MCMC)**
  Generate using a *random walk* on the witness set having polynomial time convergence to an almost uniform random solution (*rapid mixing*). First proposed by Broder, 1986, to count perfect matchings in dense bipartite graphs. Broder’s algorithm was proved correct by Jerrum and Sinclair, 1988. Notable successes include volume computation, permanent.
Direct Monte Carlo

The basic framework is that we wish to estimate the size $|S|$ of the hard set $S = \{y : \Phi(x, y)\}$ for given $x$. To do this, we enclose it in an easy set $S'$ which approximates $S$ and can be sampled uniformly.

Sample to get an estimate $\hat{p}$ of the proportion $p = |S|/|S'|$ of sampled points which lie in $S$. Estimate $|S|$ by $\hat{p} |S'|$.

The key to using this approach is to find a set $S'$ such that

- $S \subseteq S'$.
- $|S'|$ is easy to determine.
- $S'$ can be easily sampled.
- $|S'|/|S|$ is polynomially bounded.
The basic framework is that we wish to estimate the size $|S|$ of the hard set $S = \{y : \Phi(x, y)\}$ for given $x$. To do this, we enclose it in an easy set $S'$ which approximates $S$ and can be sampled uniformly.

Sample to get an estimate $\hat{p}$ of the proportion $p = |S|/|S'|$ of sampled points which lie in $S$. Estimate $|S|$ by $\hat{p} |S'|$.

The key to using this approach is to find a set $S'$ such that

- $S \subseteq S'$.
- $|S'|$ is easy to determine.
- $S'$ can be easily sampled.
- $|S'|/|S|$ is polynomially bounded.
The basic framework is that we wish to estimate the size $|S|$ of the hard set $S = \{y : \Phi(x, y)\}$ for given $x$. To do this, we enclose it in an easy set $S'$ which approximates $S$ and can be sampled uniformly.

Sample to get an estimate $\hat{p}$ of the proportion $p = |S|/|S'|$ of sampled points which lie in $S$. Estimate $|S|$ by $\hat{p} |S'|$.

The key to using this approach is to find a set $S'$ such that

- $S \subseteq S'$.
- $|S'|$ is easy to determine.
- $S'$ can be easily sampled.
- $|S'|/|S|$ is polynomially bounded.
Direct Monte Carlo

The basic framework is that we wish to estimate the size $|S|$ of the hard set $S = \{ y : \Phi(x, y) \}$ for given $x$. To do this, we enclose it in an easy set $S'$ which approximates $S$ and can be sampled uniformly.

Sample to get an estimate $\hat{p}$ of the proportion $p = |S|/|S'|$ of sampled points which lie in $S$. Estimate $|S|$ by $\hat{p} |S'|$.

The key to using this approach is to find a set $S'$ such that

- $S \subseteq S'$.
- $|S'|$ is easy to determine.
- $S'$ can be easily sampled.
- $|S'|/|S|$ is polynomially bounded.
Direct Monte Carlo

The basic framework is that we wish to estimate the size $|S|$ of the hard set $S = \{y : \Phi(x, y)\}$ for given $x$. To do this, we enclose it in an easy set $S'$ which approximates $S$ and can be sampled uniformly.

Sample to get an estimate $\hat{p}$ of the proportion $p = |S|/|S'|$ of sampled points which lie in $S$. Estimate $|S|$ by $\hat{p} |S'|$.

The key to using this approach is to find a set $S'$ such that

- $S \subseteq S'$.
- $|S'|$ is easy to determine.
- $S'$ can be easily sampled.
- $|S'|/|S|$ is polynomially bounded.
Karp and Luby, 1983, used the DMC approach to count solutions of DNF Boolean formulae.

A DNF formula has \( m \) clauses \( C_1, \ldots, C_m \) in \( n \) Boolean variables \( X_1, \ldots, X_n \). A clause is the conjunction of some literals \( X_j \) or \( \neg X_j \) \( (j = 1, \ldots, n) \). The formula is the disjunction of all the clauses, e.g.

\[
F = X_1 \lor (X_1 \land \neg X_2) \lor (X_1 \land X_3) \lor (X_2 \land \neg X_3).
\]

We may assign a truth value to any variable \( X_k \), 0 (false) or 1 (true). If \( y = (y_1, \ldots, y_n) \), with \( y_k \in \{0, 1\} \) for \( k = 1, \ldots, n \), we call \( y \) an assignment. The assignment \( y \) satisfies the formula \( F \) if \( F \) evaluates to 1 under \( y \).
Counting DNF solutions

Karp and Luby, 1983, used the DMC approach to count solutions of DNF Boolean formulae.

A **DNF** formula has $m$ clauses $C_1, \ldots, C_m$ in $n$ Boolean variables $X_1, \ldots, X_n$. A clause is the conjunction of some literals $X_j$ or $\neg X_j$ ($j = 1, \ldots, n$). The **formula** is the disjunction of all the clauses, e.g.

$$F = X_1 \vee (X_1 \land \neg X_2) \vee (X_1 \land X_3) \vee (X_2 \land \neg X_3).$$

We may assign a **truth value** to any variable $X_k$, 0 (*false*) or 1 (*true*). If $y = (y_1, \ldots, y_n)$, with $y_k \in \{0, 1\}$ for $k = 1, \ldots, n$, we call $y$ an **assignment**. The assignment $y$ **satisfies** the formula $F$ if $F$ evaluates to 1 under $y$. 
Karp and Luby, 1983, used the DMC approach to count solutions of DNF Boolean formulae.

A DNF formula has \( m \) clauses \( C_1, \ldots, C_m \) in \( n \) Boolean variables \( X_1, \ldots, X_n \). A clause is the conjunction of some literals \( X_j \) or \( \neg X_j \) \((j = 1, \ldots, n)\). The formula is the disjunction of all the clauses, e.g.

\[
F = X_1 \lor (X_1 \land \neg X_2) \lor (X_1 \land X_3) \lor (X_2 \land \neg X_3).
\]

We may assign a truth value to any variable \( X_k \), \( 0 \) (false) or \( 1 \) (true). If \( y = (y_1, \ldots, y_n) \), with \( y_k \in \{0, 1\} \) for \( k = 1, \ldots, n \), we call \( y \) an assignment. The assignment \( y \) satisfies the formula \( F \) if \( F \) evaluates to 1 under \( y \).
The Karp-Luby algorithm

The satisfying assignments of $C_i$ define a subset $S_i$ of $\{0, 1\}^n$ for $i = 1, \ldots, m$. The set of satisfying assignments of $F$ is then $S = \bigcup_{i=1}^m S_i$. We wish to estimate the $\#P$-complete quantity $|S|$. We clearly have $|S_i| = 2^n - |C_i|$ for $i = 1, \ldots, m$ and $0 \leq |S| \leq 2^n$.

We choose $S'$ to be the set of pairs $(y, i)$, where $y \in S_i$. Note then that $|S'| = \sum_{i=1}^m |S_i| \leq m|S|$.

We choose $i$ from $\{1, \ldots, m\}$ with probability $p_i = |S_i|/|S'|$, and then a uniform random $y \in S_i$, by choosing values for the unset variables in $C_i$ with probability $\frac{1}{2}$. We then determine $i^* = \min\{k : y \in S_k\}$. If $i^* = i$, we accept $y$, otherwise reject it.
The Karp-Luby algorithm

The satisfying assignments of $C_i$ define a subset $S_i$ of $\{0, 1\}^n$ for $i = 1, \ldots, m$. The set of satisfying assignments of $F$ is then $S = \bigcup_{i=1}^{m} S_i$. We wish to estimate the \#P-complete quantity $|S|$. We clearly have $|S_i| = 2^n - |C_i|$ for $i = 1, \ldots, m$ and $0 \leq |S| \leq 2^n$.

We choose $S'$ to be the set of pairs $(y, i)$, where $y \in S_i$. Note then that $|S'| = \sum_{i=1}^{m} |S_i| \leq m|S|$.

We choose $i$ from $\{1, \ldots, m\}$ with probability $p_i = |S_i|/|S'|$, and then a uniform random $y \in S_i$, by choosing values for the unset variables in $C_i$ with probability $\frac{1}{2}$. We then determine $i^* = \min\{k : y \in S_k\}$. If $i^* = i$, we accept $y$, otherwise reject it.
The Karp-Luby algorithm

The satisfying assignments of \( C_i \) define a subset \( S_i \) of \( \{0, 1\}^n \) for \( i = 1, \ldots, m \). The set of satisfying assignments of \( F \) is then \( S = \bigcup_{i=1}^{m} S_i \). We wish to estimate the \#P-complete quantity \(|S|\).

We clearly have \(|S_i| = 2^n - |C_i|\) for \( i = 1, \ldots, m \) and \( 0 \leq |S| \leq 2^n \).

We choose \( S' \) to be the set of pairs \((y, i)\), where \( y \in S_i \).

Note then that \(|S'| = \sum_{i=1}^{m} |S_i| \leq m|S|\).

We choose \( i \) from \( \{1, \ldots, m\} \) with probability \( p_i = |S_i|/|S'| \), and then a uniform random \( y \in S_i \), by choosing values for the unset variables in \( C_i \) with probability \( \frac{1}{2} \). We then determine \( i^* = \min\{k : y \in S_k\} \).

If \( i^* = i \), we accept \( y \), otherwise reject it.
The Karp-Luby algorithm

We repeat for $M$ samples $y$, for large enough $M$. Suppose we accept $K$ of these. The estimate of $|S|$ is then $|S'|K/M$.

Consider the probability that a given element $(y, i)$ of $S'$ is chosen. First the integer $i$ must be chosen. The probability of this is $|S_i|/|S'|$ by construction. Next, $y$ must be the element of $S_i$ chosen. The probability of this is $1/|S_i|$. Therefore the probability that $(y, i)$ is produced is equal to $1/|S_i| \times |S_i|/|S'| = |S'|$, so each element of $S'$ is equally likely.

There are exactly $|S|$ acceptable pairs, so the probability of an acceptance is $|S|/|S'| \geq 1/m$. It is now easy to see the estimate has the right expected value. It will have small enough variance if $M \gg m^2$. We leave the remaining details for the next example.
The Karp-Luby algorithm

We repeat for $M$ samples $y$, for large enough $M$. Suppose we accept $K$ of these. The estimate of $|S|$ is then $|S'|K/M$.

Consider the probability that a given element $(y, i)$ of $S'$ is chosen. First the integer $i$ must be chosen. The probability of this is $|S_i|/|S'|$ by construction. Next, $y$ must be the element of $S_i$ chosen. The probability of this is $1/|S_i|$. Therefore the probability that $(y, i)$ is produced is equal to $1/|S_i| \times |S_i|/|S'| = |S'|$, so each element of $S'$ is equally likely.

There are exactly $|S|$ acceptable pairs, so the probability of an acceptance is $|S|/|S'| \geq 1/m$. It is now easy to see the estimate has the right expected value. It will have small enough variance if $M \gg m^2$. We leave the remaining details for the next example.
The Karp-Luby algorithm

We repeat for $M$ samples $y$, for large enough $M$. Suppose we accept $K$ of these. The estimate of $|S|$ is then $|S'| K / M$.

Consider the probability that a given element $(y, i)$ of $S'$ is chosen. First the integer $i$ must be chosen. The probability of this is $|S_i| / |S'|$ by construction. Next, $y$ must be the element of $S_i$ chosen. The probability of this is $1 / |S_i|$. Therefore the probability that $(y, i)$ is produced is equal to $1 / |S_i| \times |S_i| / |S'| = |S'|$, so each element of $S'$ is equally likely.

There are exactly $|S|$ acceptable pairs, so the probability of an acceptance is $|S| / |S'| \geq 1 / m$. It is now easy to see the estimate has the right expected value. It will have small enough variance if $M \gg m^2$. We leave the remaining details for the next example.
Counting knapsack solutions

Dyer, 2003, used the DMC approach for the 0-1 knapsack problem.

\[ \sum_{j=1}^{n} a_j x_j \leq b, \]

where the \( a_j \geq 0 \) and \( b \) are given integers, and the \( x_j \in \{0, 1\} \) are the variables \((j = 1, 2, \ldots, n)\).

Morris and Sinclair, 1999, proved polynomial time convergence of a natural random walk on knapsack solutions. This gave the first fpras. The most recent version of their proof (2002) gives sampling in time \( O(n^{4.5}) \), leading to an fpras with running time \( O(n^{5.5}/\varepsilon^2) \).

Obtaining a faster fpras using MCMC seems to be a challenge.
Counting knapsack solutions

Dyer, 2003, used the DMC approach for the 0-1 knapsack problem.

\[ \sum_{j=1}^{n} a_j x_j \leq b, \]

where the \( a_j \geq 0 \) and \( b \) are given integers, and the \( x_j \in \{0, 1\} \) are the variables \( (j = 1, 2, \ldots, n) \).

Morris and Sinclair, 1999. proved polynomial time convergence of a natural random walk on knapsack solutions. This gave the first fpras. The most recent version of their proof (2002) gives sampling in time \( O(n^{4.5}) \), leading to an fpras with running time \( O(n^{5.5} / \varepsilon^2) \).

Obtaining a faster fpras using MCMC seems to be a challenge.
Counting knapsack solutions

Dyer, 2003, used the DMC approach for the 0-1 knapsack problem.

\[ \sum_{j=1}^{n} a_j x_j \leq b, \]

where the \( a_j \geq 0 \) and \( b \) are given integers, and the \( x_j \in \{0, 1\} \) are the variables \( (j = 1, 2, \ldots, n) \).

Morris and Sinclair, 1999. proved polynomial time convergence of a natural random walk on knapsack solutions. This gave the first fpras. The most recent version of their proof (2002) gives sampling in time \( O(n^{4.5}) \), leading to an fpras with running time \( O(n^{5.5}/\varepsilon^2) \).

Obtaining a faster fpras using MCMC seems to be a challenge.
Dart throwing revisited

Now \( S = \{ x \in \{0, 1\}^n : \sum_{j=1}^{n} a_j x_j \leq b \} \), where we assume without loss that \( 0 \leq a_1 \leq a_2 \leq \cdots \leq a_n \leq b \).

Here we will use dynamic programming to allow us to define and work with a suitable \( S' \).

We scale and round the \( a_j \) by \( \alpha_j = \lfloor n^2 a_j / b \rfloor \) \( (j = 1, 2, \ldots, n) \).

Now consider

\[
S' = \{ x \in \{0, 1\}^n : \sum_{j=1}^{n} \alpha_j x_j \leq n^2 \}.
\]

Now \( |S'| \) can be computed by dynamic programming in \( O(n^3) \) time:

Let \( F(r, s) = |\{ x \in \{0, 1\}^r : \sum_{j=1}^{r} \alpha_j x_j \leq s \}| \). Then \( |S'| = F(n, n^2) \).
Dynamic Programming

We can compute $F(n, n^2)$ in $O(n^3)$ time using the recurrence

$$F(r, s) = F(r - 1, s) + F(r - 1, s - \alpha_r) \quad (r > 1)$$

$$F(1, s) = \begin{cases} 
1 & (s < \alpha_1), \\
2 & (s \geq \alpha_1).
\end{cases}$$

We must now establish how is $|S'|$ related to $|S|$. 
Let $k$ be the largest $j$ such that $a_j \leq b/n$. We have

(1) $x \in S$ implies $x \in S'$, since then $\frac{n^2}{b} \sum_{j=1}^{n} a_j x_j \leq n^2$,

implying $\sum_{j=1}^{n} a_j x_j \leq n^2$.

(2) $x \in S'$ and $x_j = 0$ ($\forall j > k$) implies $x \in S$, since then

$\sum_{j=1}^{n} a_j x_j = \sum_{j=1}^{k} a_j x_j \leq \frac{kb}{n} \leq b$.

From (1) we have $S \subseteq S'$, so $|S| \leq |S'|$.

From (2) we see that, if $x \in S' \setminus S$, there exists a position $p(x) > k$ such that $x_p = 1$. 

Martin Dyer (Leeds University)
Let \( k \) be the largest \( j \) such that \( a_j \leq b/n \). We have

1. \( x \in S \) implies \( x \in S' \), since then\[
\frac{n^2}{b} \sum_{j=1}^{n} a_j x_j \leq n^2,
\]
   implying \( \sum_{j=1}^{n} \alpha_j x_j \leq n^2 \).

2. \( x \in S' \) and \( x_j = 0 \) (\( \forall j > k \)) implies \( x \in S \), since then
   \[
   \sum_{j=1}^{n} a_j x_j = \sum_{j=1}^{k} a_j x_j \leq \frac{kb}{n} \leq b.
   \]

From (1) we have \( S \subseteq S' \), so \( |S| \leq |S'| \).

From (2) we see that, if \( x \in S' \setminus S \), there exists a position \( p(x) > k \) such that \( x_p = 1 \).
Let $k$ be the largest $j$ such that $a_j \leq b/n$. We have

(1) $x \in S$ implies $x \in S'$, since then $\frac{n^2}{b} \sum_{j=1}^{n} a_j x_j \leq n^2$,

implying $\sum_{j=1}^{n} \alpha_j x_j \leq n^2$.

(2) $x \in S'$ and $x_j = 0$ (\forall j > k) implies $x \in S$, since then

$$\sum_{j=1}^{n} a_j x_j = \sum_{j=1}^{k} a_j x_j \leq \frac{kb}{n} \leq b.$$

From (1) we have $S \subseteq S'$, so $|S| \leq |S'|$.

From (2) we see that, if $x \in S' \setminus S$, there exists a position $p(x) > k$ such that $x_p = 1$. 
Let $k$ be the largest $j$ such that $a_j \leq b/n$. We have

(1) $x \in S$ implies $x \in S'$, since then $\frac{n^2}{b} \sum_{j=1}^{n} a_jx_j \leq n^2$,

implying $\sum_{j=1}^{n} a_jx_j \leq n^2$.

(2) $x \in S'$ and $x_j = 0 \ (\forall j > k)$ implies $x \in S$, since then

$\sum_{j=1}^{n} a_jx_j = \sum_{j=1}^{k} a_jx_j \leq \frac{kb}{n} \leq b$.

From (1) we have $S \subseteq S'$, so $|S| \leq |S'|$.

From (2) we see that, if $x \in S' \setminus S$, there exists a position $p(x) > k$ such that $x_p = 1$. 
Note that \( \alpha_p = \left\lfloor \frac{n^2 a_p}{b} \right\rfloor \geq \left\lfloor \frac{n^2(b/n)}{b} \right\rfloor \geq n \).

Consider a mapping \( f : S' \rightarrow \{0, 1\}^n \) defined by

(i) \( f(x) = x \) if \( x \in S \),

(ii) \( f(x) = y \) if \( x \in S' \setminus S \) where \( y_j = \begin{cases} 0 & (j = p) \\ x_j & (j \neq p) \end{cases} \).

(If more than one such \( p(x) \) exists, choose the largest.)

We can now show easily that \( f(S') = S \).

Case (i) is trivial, so suppose we have case (ii), i.e. \( x \in S' \setminus S \). Then \( p(x) \) is well defined.
Note that $\alpha_p = \left\lfloor \frac{n^2 a_p}{b} \right\rfloor \geq \left\lfloor \frac{n^2 (b/n)}{b} \right\rfloor \geq n$.

Consider a mapping $f : S' \to \{0, 1\}^n$ defined by

(i) $f(x) = x$ if $x \in S$,

(ii) $f(x) = y$ if $x \in S' \setminus S$ where $y_j = \begin{cases} 0 & (j = p) \\ x_j & (j \neq p) \end{cases}$.

(If more than one such $p(x)$ exists, choose the largest.)

We can now show easily that $f(S') = S$.

Case (i) is trivial, so suppose we have case (ii), i.e. $x \in S' \setminus S$. Then $p(x)$ is well defined.
Note that $\alpha_p = \left\lfloor \frac{n^2 a_p}{b} \right\rfloor \geq \left\lfloor \frac{n^2 (b/n)}{b} \right\rfloor \geq n$.

Consider a mapping $f : S' \to \{0, 1\}^n$ defined by

(i) $f(x) = x$ if $x \in S$,

(ii) $f(x) = y$ if $x \in S' \setminus S$ where $y_j = \begin{cases} 0 & (j = p) \\ x_j & (j \neq p) \end{cases}$.

(If more than one such $p(x)$ exists, choose the largest.)

We can now show easily that $f(S') = S$.

Case (i) is trivial, so suppose we have case (ii), i.e. $x \in S' \setminus S$. Then $p(x)$ is well defined.
Note that $\alpha_p = \left\lfloor \frac{n^2a_p}{b} \right\rfloor \geq \left\lfloor \frac{n^2(b/n)}{b} \right\rfloor \geq n$.

Consider a mapping $f : S' \rightarrow \{0, 1\}^n$ defined by

(i) $f(x) = x$ if $x \in S$,

(ii) $f(x) = y$ if $x \in S' \setminus S$ where $y_j = \begin{cases} 0 & (j = p) \\ x_j & (j \neq p) \end{cases}$.

(If more than one such $p(x)$ exists, choose the largest.)

We can now show easily that $f(S') = S$.

Case (i) is trivial, so suppose we have case (ii), i.e. $x \in S' \setminus S$. Then $p(x)$ is well defined.
Note that $\alpha_p = \left\lfloor \frac{n^2 a_p}{b} \right\rfloor \geq \left\lfloor \frac{n^2 (b/n)}{b} \right\rfloor \geq n$.

Consider a mapping $f : S' \rightarrow \{0, 1\}^n$ defined by

(i) $f(x) = x$ if $x \in S$,

(ii) $f(x) = y$ if $x \in S' \setminus S$ where $y_j = \begin{cases} 0 & (j = p) \\ x_j & (j \neq p) \end{cases}$.

(If more than one such $p(x)$ exists, choose the largest.)

We can now show easily that $f(S') = S$.

Case (i) is trivial, so suppose we have case (ii), i.e. $x \in S' \setminus S$. Then $p(x)$ is well defined.
Note that \( \alpha_p = \left\lfloor \frac{n^2 a_p}{b} \right\rfloor \geq \left\lfloor \frac{n^2 (b/n)}{b} \right\rfloor \geq n \).

Consider a mapping \( f : S' \rightarrow \{0, 1\}^n \) defined by

(i) \( f(x) = x \) if \( x \in S \),

(ii) \( f(x) = y \) if \( x \in S' \setminus S \) where \( y_j = \begin{cases} 0 & (j = p) \\ x_j & (j \neq p) \end{cases} \).

(If more than one such \( p(x) \) exists, choose the largest.)

We can now show easily that \( f(S') = S \).

Case (i) is trivial, so suppose we have case (ii), i.e. \( x \in S' \setminus S \). Then \( p(x) \) is well defined.
Analysis (3)

Let \( y = f(x) \). Then \( y \in S \) follows from

\[
\sum_{j=1}^{n} a_j y_j \leq \frac{b}{n^2} \sum_{j=1}^{n} (\alpha_j + 1) y_j \leq \frac{b}{n^2} \left( \sum_{j=1}^{n} \alpha_j y_j + n \right)
\]

\[
= \frac{b}{n^2} \left( \sum_{j=1}^{n} \alpha_j x_j - \alpha_p + n \right) \leq \frac{b}{n^2} \sum_{j=1}^{n} \alpha_j x_j \quad \text{(since } \alpha_p \geq n),
\]

\[
\leq b \quad \text{(since } x \in S'),
\]

Hence \( f(S') = S \), as claimed.

For any \( y \in S \), \( |f^{-1}(y)| \leq n + 1 \), since we may have \( f(x) = y \) for \( x = y \) and for any \( x \) which comes from changing one 0 in \( y \) to a 1.

Thus we have shown \( |S'| \leq (n+1)|S| \).
Let \( y = f(x) \). Then \( y \in S \) follows from

\[
\sum_{j=1}^{n} a_j y_j \leq \frac{b}{n^2} \sum_{j=1}^{n} (\alpha_j + 1)y_j \leq \frac{b}{n^2} \left( \sum_{j=1}^{n} \alpha_j y_j + n \right)
\]

\[
= \frac{b}{n^2} \left( \sum_{j=1}^{n} \alpha_j x_j - \alpha_p + n \right) \leq \frac{b}{n^2} \sum_{j=1}^{n} \alpha_j x_j \quad \text{(since } \alpha_p \geq n),
\]

\[
\leq b \quad \text{(since } x \in S').
\]

Hence \( f(S') = S \), as claimed.

For any \( y \in S \), \( |f^{-1}(y)| \leq n + 1 \), since we may have \( f(x) = y \) for \( x = y \) and for any \( x \) which comes from changing one 0 in \( y \) to a 1.

Thus we have shown \( |S'| \leq (n + 1)|S| \).
Let $y = f(x)$. Then $y \in S$ follows from

$$\sum_{j=1}^{n} a_j y_j \leq b \frac{n}{n^2} \sum_{j=1}^{n} (\alpha_j + 1) y_j \leq b \frac{n}{n^2} \left( \sum_{j=1}^{n} \alpha_j y_j + n \right)$$

$$= b \frac{n}{n^2} \left( \sum_{j=1}^{n} \alpha_j x_j - \alpha_P + n \right) \leq b \frac{n}{n^2} \sum_{j=1}^{n} \alpha_j x_j \quad \text{(since } \alpha_P \geq n),$$

$$\leq b \quad \text{(since } x \in S').$$

Hence $f(S') = S$, as claimed.

For any $y \in S$, $|f^{-1}(y)| \leq n + 1$, since we may have $f(x) = y$ for $x = y$ and for any $x$ which comes from changing one 0 in $y$ to a 1.

Thus we have shown $|S'| \leq (n + 1)|S|$. 
Let $y = f(x)$. Then $y \in S$ follows from

$$\sum_{j=1}^{n} a_j y_j \leq \frac{b}{n^2} \sum_{j=1}^{n} (\alpha_j + 1) y_j \leq \frac{b}{n^2} \left( \sum_{j=1}^{n} \alpha_j y_j + n \right)$$

$$= \frac{b}{n^2} \left( \sum_{j=1}^{n} \alpha_j x_j - \alpha_p + n \right) \leq \frac{b}{n^2} \sum_{j=1}^{n} \alpha_j x_j \quad \text{(since } \alpha_p \geq n\text{)},$$

$$\leq b \quad \text{(since } x \in S'\text{)}.$$ 

Hence $f(S') = S$, as claimed.

For any $y \in S$, $|f^{-1}(y)| \leq n + 1$, since we may have $f(x) = y$ for $x = y$ and for any $x$ which comes from changing one 0 in $y$ to a 1.

Thus we have shown $|S'| \leq (n + 1)|S|$. 
Let $y = f(x)$. Then $y \in S$ follows from

$$\sum_{j=1}^{n} a_j y_j \leq \frac{b}{n^2} \sum_{j=1}^{n} (\alpha_j + 1)y_j \leq \frac{b}{n^2} \left( \sum_{j=1}^{n} \alpha_j y_j + n \right)$$

$$= \frac{b}{n^2} \left( \sum_{j=1}^{n} \alpha_j x_j - \alpha_p + n \right) \leq \frac{b}{n^2} \sum_{j=1}^{n} \alpha_j x_j \quad \text{(since } \alpha_p \geq n),$$

$$\leq b \quad \text{(since } x \in S').$$

Hence $f(S') = S$, as claimed.

For any $y \in S$, $|f^{-1}(y)| \leq n + 1$, since we may have $f(x) = y$ for $x = y$ and for any $x$ which comes from changing one 0 in $y$ to a 1.

Thus we have shown $|S'| \leq (n + 1)|S|$. 

Hence $1 \leq \frac{|S'|}{|S|} \leq n + 1$, so $\frac{|S'|}{\sqrt{n+1}}$ approximates $|S|$ deterministically to within a factor $\sqrt{n + 1}$, and is computable in time $O(n^3)$.

This implies almost uniform sampling from $S$ can be done in polynomial time, and the existence of an fpras, by applying a general MCMC technique of Sinclair and Jerrum, 1989.

But there is a simpler (and more efficient) way of obtaining exact uniform sampling and an fpras. In fact, the dynamic programming tabulation gives much more information than simply $|S'|$. 
Sampling

Hence $1 \leq \frac{|S'|}{|S|} \leq n + 1$, so $\frac{|S'|}{\sqrt{n+1}}$ approximates $|S|$ deterministically to within a factor $\sqrt{n+1}$, and is computable in time $O(n^3)$.

This implies almost uniform sampling from $S$ can be done in polynomial time, and the existence of an fpras, by applying a general MCMC technique of Sinclair and Jerrum, 1989.

But there is a simpler (and more efficient) way of obtaining exact uniform sampling and an fpras. In fact, the dynamic programming tabulation gives much more information than simply $|S'|$. 
Hence \(1 \leq \frac{|S'|}{|S|} \leq n + 1\), so \(\frac{|S'|}{\sqrt{n+1}}\) approximates \(|S|\) deterministically to within a factor \(\sqrt{n + 1}\), and is computable in time \(O(n^3)\).

This implies almost uniform sampling from \(S\) can be done in polynomial time, and the existence of an fpras, by applying a general MCMC technique of Sinclair and Jerrum, 1989.

But there is a simpler (and more efficient) way of obtaining exact uniform sampling and an fpras. In fact, the dynamic programming tabulation gives much more information than simply \(|S'|\).
We have \( \Pr(x_n = 0) = \frac{F(n-1, n^2)}{F(n, n^2)} \), \( \Pr(x_n = 1) = \frac{F(n-1, n^2 - \alpha_n)}{F(n, n^2)} \),

Hence we can flip a biased coin to choose \( x_n \). Similarly, we can select \( x_{n-1} \) once \( x_n \) is chosen, and hence trace back to get a uniform \( x \in S' \) in only \( O(n) \) time.
We have $\Pr(x_n = 0) = \frac{F(n-1, n^2)}{F(n, n^2)}$, $\Pr(x_n = 1) = \frac{F(n-1, n^2 - \alpha_n)}{F(n, n^2)}$.

Hence we can flip a biased coin to choose $x_n$. Similarly, we can select $x_{n-1}$ once $x_n$ is chosen, and hence trace back to get a uniform $x \in S'$ in only $O(n)$ time.
Since \( \Pr(x \in S | x \in S') = \frac{|S|}{|S'|} \geq \frac{1}{n+1} \) we need \( O(n) \) trials to get a single uniform point \( x \in S \). To get a sample of size \( \nu \), we therefore need only \( O(n^3 + \nu n^2) \) time, with very high probability (Chernoff).

The fpras is equally easy. Estimate \( \rho = \frac{|S|}{|S'|} \) by sampling \( x \in S' \) and checking if \( x \in S \) for a large enough sample.

The sampling error in the estimate \( \hat{\rho} \) of \( \rho \) is \( O(\sqrt{\rho/\nu}) = O(1/\sqrt{n\nu}) \) with sample size \( \nu \). To get within \( \varepsilon \rho = \Omega(\varepsilon/n) \) of \( \rho \), we need only \( \nu = O(n/\varepsilon^2) \). (High probability by Chernoff.) The fpras has running time \( O(n^3 + n^2/\varepsilon^2) \).

This can be improved to \( O(n^{5/2} \sqrt{\log(n/\varepsilon)} + n^2/\varepsilon^2) \) using randomized rounding, rather than always rounding down.
Analysis (4)

Since \( \Pr(x \in S \mid x \in S') = \frac{|S|}{|S'|} \geq \frac{1}{n+1} \) we need \( O(n) \) trials to get a single uniform point \( x \in S \). To get a sample of size \( \nu \), we therefore need only \( O(n^3 + \nu n^2) \) time, with very high probability (Chernoff).

The fpras is equally easy. Estimate \( \rho = \frac{|S|}{|S'|} \) by sampling \( x \in S' \) and checking if \( x \in S \) for a large enough sample.

The sampling error in the estimate \( \hat{\rho} \) of \( \rho \) is \( O(\sqrt{\frac{\rho}{\nu}}) = O(1/\sqrt{\nu}) \) with sample size \( \nu \). To get within \( \epsilon \rho = \Omega(\epsilon/n) \) of \( \rho \), we need only \( \nu = O(n/\epsilon^2) \). (High probability by Chernoff.) The fpras has running time \( O(n^3 + n^2/\epsilon^2) \).

This can be improved to \( O(n^{5/2} \sqrt{\log(n/\epsilon)} + n^2/\epsilon^2) \) using randomized rounding, rather than always rounding down.
Since \( \Pr(x \in S \mid x \in S') = \frac{|S|}{|S'|} \geq \frac{1}{n+1} \) we need \( O(n) \) trials to get a single uniform point \( x \in S \). To get a sample of size \( \nu \), we therefore need only \( O(n^3 + \nu n^2) \) time, with very high probability (Chernoff).

The fpras is equally easy. Estimate \( \rho = \frac{|S|}{|S'|} \) by sampling \( x \in S' \) and checking if \( x \in S \) for a large enough sample.

The sampling error in the estimate \( \hat{\rho} \) of \( \rho \) is \( O(\sqrt{\rho/\nu}) = O(1/\sqrt{n\nu}) \) with sample size \( \nu \). To get within \( \varepsilon \rho = \Omega(\varepsilon/n) \) of \( \rho \), we need only \( \nu = O(n/\varepsilon^2) \). (High probability by Chernoff.) The fpras has running time \( O(n^3 + n^2/\varepsilon^2) \).

This can be improved to \( O(n^{5/2} \sqrt{\log(n/\varepsilon)} + n^2/\varepsilon^2) \) using randomized rounding, rather than always rounding down.
Since $\Pr(x \in S \mid x \in S') = \frac{|S|}{|S'|} \geq \frac{1}{n+1}$ we need $O(n)$ trials to get a single uniform point $x \in S$. To get a sample of size $\nu$, we therefore need only $O(n^3 + \nu n^2)$ time, with very high probability (Chernoff).

The fpras is equally easy. Estimate $\rho = \frac{|S|}{|S'|}$ by sampling $x \in S'$ and checking if $x \in S$ for a large enough sample.

The sampling error in the estimate $\hat{\rho}$ of $\rho$ is $O(\sqrt{\rho/\nu}) = O(1/\sqrt{n\nu})$ with sample size $\nu$. To get within $\varepsilon \rho = \Omega(\varepsilon/n)$ of $\rho$, we need only $\nu = O(n/\varepsilon^2)$. (High probability by Chernoff.) The fpras has running time $O(n^3 + n^2/\varepsilon^2)$.

This can be improved to $O(n^{5/2} \sqrt{\log(n/\varepsilon)} + n^2/\varepsilon^2)$ using randomized rounding, rather than always rounding down.
Extensions

The method we have described can be extended to

(1) General integer knapsack, where \( 0 \leq x_j \leq u_j (j = 1, 2, \ldots, n) \), for given integers \( u_j \). The rounded problem is now nonlinear.

(2) Multidimensional knapsack, when the number of constraints \( m \) is a constant. The previous best fpras for this problem had time complexity \( n^{2O(m)} / \varepsilon^2 \). (Morris and Sinclair, 2000)

(3) Contingency tables, when the number of rows \( m \) is a constant. (Constructing \( f \) here is trickier.) The previous best fpras for this problem had time complexity \( n^{O(m^2)} / \varepsilon^2 \). (Cryan and Dyer, 2002)

For all these problems, we get an fpras with complexity \( n^{O(m)} / \varepsilon^2 \), an improvement for (2) and (3). Problem (1) had no previous fpras.
The method we have described can be extended to

(1) General integer knapsack, where $0 \leq x_j \leq u_j$ ($j = 1, 2, \ldots, n$), for given integers $u_j$. The rounded problem is now nonlinear.

(2) Multidimensional knapsack, when the number of constraints $m$ is a constant. The previous best fpras for this problem had time complexity $n^{2O(m)}/\varepsilon^2$. (Morris and Sinclair, 2000)

(3) Contingency tables, when the number of rows $m$ is a constant. (Constructing $f$ here is trickier.) The previous best fpras for this problem had time complexity $n^{O(m^2)}/\varepsilon^2$. (Cryan and Dyer, 2002)

For all these problems, we get an fpras with complexity $n^{O(m)}/\varepsilon^2$, an improvement for (2) and (3). Problem (1) had no previous fpras.
Extensions

The method we have described can be extended to

(1) General integer knapsack, where \( 0 \leq x_j \leq u_j \) \((j = 1, 2, \ldots, n)\), for given integers \( u_j \). The rounded problem is now nonlinear.

(2) Multidimensional knapsack, when the number of constraints \( m \) is a constant. The previous best fpras for this problem had time complexity \( n^{2O(m)} / \varepsilon^2 \). (Morris and Sinclair, 2000)

(3) Contingency tables, when the number of rows \( m \) is a constant. (Constructing \( f \) here is trickier.) The previous best fpras for this problem had time complexity \( n^{O(m^2)} / \varepsilon^2 \). (Cryan and Dyer, 2002)

For all these problems, we get an fpras with complexity \( n^{O(m)} / \varepsilon^2 \), an improvement for (2) and (3). Problem (1) had no previous fpras.
Extensions

The method we have described can be extended to

(1) General integer knapsack, where $0 \leq x_j \leq u_j \ (j = 1, 2, \ldots, n)$, for given integers $u_j$. The rounded problem is now nonlinear.

(2) Multidimensional knapsack, when the number of constraints $m$ is a constant. The previous best fpras for this problem had time complexity $n^{2^{O(m)}/\varepsilon^2}$. (Morris and Sinclair, 2000)

(3) Contingency tables, when the number of rows $m$ is a constant. (Constructing $f$ here is trickier.) The previous best fpras for this problem had time complexity $n^{O(m^2)/\varepsilon^2}$. (Cryan and Dyer, 2002)

For all these problems, we get an fpras with complexity $n^{O(m)/\varepsilon^2}$, an improvement for (2) and (3). Problem (1) had no previous fpras.
Markov chain Monte Carlo is more ambitious. We try to sample from the hard set $S$ without using an easy envelope $S'$. We do this by performing a random walk on the elements of $S$, and proving that this will approach the uniform distribution in time polynomial in the instance size $|x|$.

Generally, it is not too difficult to design a random walk which reaches the uniform distribution asymptotically. The difficulty is in proving the polynomial time convergence property, even supposing it is true. In fact, sometimes we can, prove that the random walk requires exponential time to approach the uniform distribution on $S$.

This means that the method is not as easy to apply rigorously as it may appear at first sight. It is used by practitioners without proof.
Markov chain Monte Carlo is more ambitious. We try to sample from the hard set $S$ without using an easy envelope $S'$. We do this by performing a random walk on the elements of $S$, and proving that this will approach the uniform distribution in time polynomial in the instance size $|x|$. Generally, it is not too difficult to design a random walk which reaches the uniform distribution asymptotically. The difficulty is in proving the polynomial time convergence property, even supposing it is true. In fact, sometimes we can, prove that the random walk requires exponential time to approach the uniform distribution on $S$. This means that the method is not as easy to apply rigorously as it may appear at first sight. It is used by practitioners without proof.
Markov chain Monte Carlo is more ambitious. We try to sample from the hard set $S$ without using an easy envelope $S'$. We do this by performing a random walk on the elements of $S$, and proving that this will approach the uniform distribution in time polynomial in the instance size $|x|$.

Generally, it is not too difficult to design a random walk which reaches the uniform distribution asymptotically. The difficulty is in proving the polynomial time convergence property, even supposing it is true. In fact, sometimes we can, prove that the random walk requires exponential time to approach the uniform distribution on $S$.

This means that the method is not as easy to apply rigorously as it may appear at first sight. It is used by practitioners without proof.
Example: Independent sets

Suppose $S$ is the set of all independent sets in a graph $G$ of maximum degree $\Delta$. Consider the following random walk on $S$:

Starting from any independent set $I$, repeat:

(a) Pick a vertex $u$ uniformly at random.
(b) If $u \in I$, delete it from $I$ with probability $\frac{1}{2}$.
(c) If $u \notin I$, insert it into $I$ with probability $\frac{1}{2}$, if possible.

After sufficiently many steps, this process generates an almost uniformly random independent set $I$. 
Example: Independent sets

Suppose \( S \) is the set of all independent sets in a graph \( G \) of maximum degree \( \Delta \). Consider the following random walk on \( S \):

Starting from any independent set \( I \), repeat:

(a) Pick a vertex \( u \) uniformly at random.
(b) If \( u \in I \), delete it from \( I \) with probability \( \frac{1}{2} \).
(c) If \( u \notin I \), insert it into \( I \) with probability \( \frac{1}{2} \), if possible.

After sufficiently many steps, this process generates an almost uniformly random independent set \( I \).
Example: Independent sets

Suppose $S$ is the set of all independent sets in a graph $G$ of maximum degree $\Delta$. Consider the following random walk on $S$:

Starting from any independent set $I$, repeat:

(a) Pick a vertex $u$ uniformly at random.
(b) If $u \in I$, delete it from $I$ with probability $\frac{1}{2}$.
(c) If $u \notin I$, insert it into $I$ with probability $\frac{1}{2}$, if possible.

After sufficiently many steps, this process generates an almost uniformly random independent set $I$. 
Example: Independent sets

Suppose $S$ is the set of all independent sets in a graph $G$ of maximum degree $\Delta$. Consider the following random walk on $S$:

Starting from any independent set $I$, repeat:

(a) Pick a vertex $u$ uniformly at random.
(b) If $u \in I$, delete it from $I$ with probability $\frac{1}{2}$.
(c) If $u \notin I$, insert it into $I$ with probability $\frac{1}{2}$, if possible.

After sufficiently many steps, this process generates an almost uniformly random independent set $I$. 
The appropriate formalisation is the theory of Markov chains:

A Markov chain \( M = (X_t) \) on a finite state space \( \Omega \) is determined by a transition matrix \( P \) such that

\[
P(\omega, \omega') = \Pr(X_{t+1} = \omega' \mid X_t = \omega) \quad (\omega, \omega' \in \Omega).
\]

If the initial state is \( X_0 = \omega_0 \), the distribution of \( X_t \) is \( p_t(\omega) = P^t(\omega_0, \omega) \).

If \( M \) is ergodic, there is a unique equilibrium distribution \( \pi \) such that

\[
\lim_{t \to \infty} p_t = \pi \quad \text{and} \quad \pi P = \pi.
\]

Convergence to \( \pi \) is usually measured by total variation distance

\[
d_{TV}(p_t, \pi) = \frac{1}{2} \sum_{\omega \in \Omega} |p_t(\omega) - \pi(\omega)|.
\]
Markov chains

The appropriate formalisation is the theory of Markov chains:

A Markov chain $\mathcal{M} = (X_t)$ on a finite state space $\Omega$ is determined by a transition matrix $P$ such that

$$P(\omega, \omega') = \Pr(X_{t+1} = \omega' \mid X_t = \omega) \quad (\omega, \omega' \in \Omega).$$

If the initial state is $X_0 = \omega_0$, the distribution of $X_t$ is $p_t(\omega) = P^t(\omega_0, \omega)$.

If $\mathcal{M}$ is ergodic, there is a unique equilibrium distribution $\pi$ such that

$$\lim_{t \to \infty} p_t = \pi \quad \text{and} \quad \pi P = \pi.$$

Convergence to $\pi$ is usually measured by total variation distance

$$d_{TV}(p_t, \pi) = \frac{1}{2} \sum_{\omega \in \Omega} \left| p_t(\omega) - \pi(\omega) \right|.$$
The appropriate formalisation is the theory of Markov chains:

A Markov chain $\mathcal{M} = (X_t)$ on a finite state space $\Omega$ is determined by a transition matrix $P$ such that

$$P(\omega, \omega') = \Pr(X_{t+1} = \omega' \mid X_t = \omega) \quad (\omega, \omega' \in \Omega).$$

If the initial state is $X_0 = \omega_0$, the distribution of $X_t$ is $p_t(\omega) = P^t(\omega_0, \omega)$.

If $\mathcal{M}$ is ergodic, there is a unique equilibrium distribution $\pi$ such that

$$\lim_{t \to \infty} p_t = \pi \quad \text{and} \quad \pi P = \pi.$$

Convergence to $\pi$ is usually measured by total variation distance

$$d_{TV}(p_t, \pi) = \frac{1}{2} \sum_{\omega \in \Omega} |p_t(\omega) - \pi(\omega)|.$$
Markov chains

The appropriate formalisation is the theory of Markov chains:

A Markov chain \( \mathcal{M} = (X_t) \) on a finite state space \( \Omega \) is determined by a transition matrix \( P \) such that

\[
P(\omega, \omega') = \Pr(X_{t+1} = \omega' \mid X_t = \omega) \quad (\omega, \omega' \in \Omega).
\]

If the initial state is \( X_0 = \omega_0 \), the distribution of \( X_t \) is \( p_t(\omega) = P^t(\omega_0, \omega) \).

If \( \mathcal{M} \) is ergodic, there is a unique equilibrium distribution \( \pi \) such that

\[
\lim_{t \to \infty} p_t = \pi \quad \text{and} \quad \pi P = \pi.
\]

Convergence to \( \pi \) is usually measured by total variation distance

\[
d_{TV}(p_t, \pi) = \frac{1}{2} \sum_{\omega \in \Omega} |p_t(\omega) - \pi(\omega)|.
\]
Markov chains

The appropriate formalisation is the theory of Markov chains:

A Markov chain \( \mathcal{M} = (X_t) \) on a finite state space \( \Omega \) is determined by a transition matrix \( P \) such that

\[
P(\omega, \omega') = \Pr(X_{t+1} = \omega' | X_t = \omega) \quad (\omega, \omega' \in \Omega).
\]

If the initial state is \( X_0 = \omega_0 \), the distribution of \( X_t \) is \( p_t(\omega) = P^t(\omega_0, \omega) \).

If \( \mathcal{M} \) is ergodic, there is a unique equilibrium distribution \( \pi \) such that

\[
\lim_{t \to \infty} p_t = \pi \quad \text{and} \quad \pi P = \pi.
\]

Convergence to \( \pi \) is usually measured by total variation distance

\[
d_{TV}(p_t, \pi) = \frac{1}{2} \sum_{\omega \in \Omega} |p_t(\omega) - \pi(\omega)|.
\]
Determining the equilibrium distribution $\pi$ of a large Markov chain may be difficult unless we have time reversibility, i.e. $P$ satisfies

$$
\pi(\omega)P(\omega, \omega') = \pi(\omega')P(\omega', \omega) \quad (\omega, \omega' \in \Omega).
$$

These are called the detailed balance equations. Fortunately many Markov chains of interest have this property, often by design.

Conversely, if any positive vector satisfies the detailed balance equations, it is easy to show it is proportional to $\pi$.

A particular case is where $P$ is symmetric,

$$
P(\omega, \omega') = P(\omega', \omega) \quad (\omega, \omega' \in \Omega).
$$

Then the equilibrium distribution $\pi$ is clearly uniform.
Determining the equilibrium distribution $\pi$ of a large Markov chain may be difficult unless we have time reversibility, i.e. $P$ satisfies

$$\pi(\omega)P(\omega, \omega') = \pi(\omega')P(\omega', \omega) \quad (\omega, \omega' \in \Omega).$$

These are called the detailed balance equations. Fortunately many Markov chains of interest have this property, often by design.

Conversely, if any positive vector satisfies the detailed balance equations, it is easy to show it is proportional to $\pi$.

A particular case is where $P$ is symmetric,

$$P(\omega, \omega') = P(\omega', \omega) \quad (\omega, \omega' \in \Omega).$$

Then the equilibrium distribution $\pi$ is clearly uniform.
Equilibrium distribution

Determining the equilibrium distribution $\pi$ of a large Markov chain may be difficult unless we have time reversibility, i.e. $P$ satisfies

$$\pi(\omega)P(\omega, \omega') = \pi(\omega')P(\omega', \omega) \quad (\omega, \omega' \in \Omega).$$

These are called the detailed balance equations. Fortunately many Markov chains of interest have this property, often by design.

Conversely, if any positive vector satisfies the detailed balance equations, it is easy to show it is proportional to $\pi$.

A particular case is where $P$ is symmetric,

$$P(\omega, \omega') = P(\omega', \omega) \quad (\omega, \omega' \in \Omega).$$

Then the equilibrium distribution $\pi$ is clearly uniform.
Determining the equilibrium distribution $\pi$ of a large Markov chain may be difficult unless we have time reversibility, i.e. $P$ satisfies

$$\pi(\omega)P(\omega, \omega') = \pi(\omega')P(\omega', \omega) \quad (\omega, \omega' \in \Omega).$$

These are called the detailed balance equations. Fortunately many Markov chains of interest have this property, often by design.

Conversely, if any positive vector satisfies the detailed balance equations, it is easy to show it is proportional to $\pi$.

A particular case is where $P$ is symmetric,

$$P(\omega, \omega') = P(\omega', \omega) \quad (\omega, \omega' \in \Omega).$$

Then the equilibrium distribution $\pi$ is clearly uniform.
Example: Independent sets

The random walk on independent sets is a Markov chain on the state space $S$, the set of all independent sets in $G$.

If $G$ has $n$ vertices, and $I, I'$ are independent sets, then:

$$P(I, I') = \begin{cases} 1/2n, & \text{if } I' = I \setminus \{v\} \text{ for some } v; \\ 1/2n, & \text{if } I' = I \cup \{v\} \text{ and is independent; } \\ 0, & \text{otherwise.} \end{cases}$$

Note that $P(I, I') = P(I', I)$ for all $I, I' \in S$, i.e. the chain is symmetric.

Therefore we know that $\pi$ is the uniform distribution.

We must now consider how to show fast convergence (rapid mixing).
The random walk on independent sets is a Markov chain on the state space $S$, the set of all independent sets in $G$.

If $G$ has $n$ vertices, and $I, I'$ are independent sets, then:

$$P(I, I') = \begin{cases} 
\frac{1}{2n}, & \text{if } I' = I \setminus \{v\} \text{ for some } v; \\
\frac{1}{2n}, & \text{if } I' = I \cup \{v\} \text{ and is independent;} \\
0, & \text{otherwise.}
\end{cases}$$

Note that $P(I, I') = P(I', I)$ for all $I, I' \in S$, i.e. the chain is symmetric.

Therefore we know that $\pi$ is the uniform distribution.

We must now consider how to show fast convergence (rapid mixing).
Example: Independent sets

The random walk on independent sets is a Markov chain on the state space $S$, the set of all independent sets in $G$.

If $G$ has $n$ vertices, and $I, I'$ are independent sets, then:

$$P(I, I') = \begin{cases} 
\frac{1}{2n}, & \text{if } I' = I \setminus \{v\} \text{ for some } v; \\
\frac{1}{2n}, & \text{if } I' = I \cup \{v\} \text{ and is independent; } \\
0, & \text{otherwise.}
\end{cases}$$

Note that $P(I, I') = P(I', I)$ for all $I, I' \in S$, i.e. the chain is symmetric.

Therefore we know that $\pi$ is the uniform distribution.

We must now consider how to show fast convergence (rapid mixing).
Example: Independent sets

The random walk on independent sets is a Markov chain on the state space \( S \), the set of all independent sets in \( G \).

If \( G \) has \( n \) vertices, and \( I, I' \) are independent sets, then:

\[
P(I, I') = \begin{cases} 
\frac{1}{2n}, & \text{if } I' = I \setminus \{v\} \text{ for some } v; \\
\frac{1}{2n}, & \text{if } I' = I \cup \{v\} \text{ and is independent; } \\
0, & \text{otherwise.}
\end{cases}
\]

Note that \( P(I, I') = P(I', I) \) for all \( I, I' \in S \), i.e. the chain is symmetric.

Therefore we know that \( \pi \) is the uniform distribution.

We must now consider how to show fast convergence (rapid mixing).
Example: Independent sets

The random walk on independent sets is a Markov chain on the state space $S$, the set of all independent sets in $G$.

If $G$ has $n$ vertices, and $I, I'$ are independent sets, then:

$$P(I, I') = \begin{cases} \frac{1}{2n}, & \text{if } I' = I \setminus \{v\} \text{ for some } v; \\ \frac{1}{2n}, & \text{if } I' = I \cup \{v\} \text{ and is independent;} \\ 0, & \text{otherwise.} \end{cases}$$

Note that $P(I, I') = P(I', I)$ for all $I, I' \in S$, i.e. the chain is symmetric.

Therefore we know that $\pi$ is the uniform distribution.

We must now consider how to show fast convergence (rapid mixing).
Coupling is one way of bounding the convergence time.

A coupling for a Markov chain $\mathcal{M}$ is a random process $(X_t, Y_t)$ on $\Omega \times \Omega$ such that each of $X_t$, $Y_t$ is marginally a faithful copy of $\mathcal{M}$.

If we assume that $(Y_t)$ starts with $Y_0$ in the equilibrium distribution, the Coupling Lemma (Doeblin, 1933) states that $d_{TV}(p_t, \pi) \leq \Pr(X_t \neq Y_t)$.

To apply this, it is necessary to design a good coupling, so that quickly we are likely to have $X_t = Y_t$.

We must prove that this happens. The usual method is to use a metric $d(X_t, Y_t)$, and design the coupling to reduce it in expectation at each step.
Coupling

Coupling is one way of bounding the convergence time.

A **coupling** for a Markov chain $\mathcal{M}$ is a random process $(X_t, Y_t)$ on $\Omega \times \Omega$ such that each of $X_t$, $Y_t$ is marginally a faithful copy of $\mathcal{M}$.

If we assume that $(Y_t)$ starts with $Y_0$ in the equilibrium distribution, the **Coupling Lemma** (Doeblin, 1933) states that $d_{TV}(p_t, \pi) \leq \Pr(X_t \neq Y_t)$.

To apply this, it is necessary to design a good coupling, so that quickly we are likely to have $X_t = Y_t$.

We must prove that this happens. The usual method is to use a metric $d(X_t, Y_t)$, and design the coupling to reduce it in expectation at each step.
Coupling

Coupling is one way of bounding the convergence time.

A coupling for a Markov chain $\mathcal{M}$ is a random process $(X_t, Y_t)$ on $\Omega \times \Omega$ such that each of $X_t$, $Y_t$ is marginally a faithful copy of $\mathcal{M}$.

If we assume that $(Y_t)$ starts with $Y_0$ in the equilibrium distribution, the Coupling Lemma (Doeblin, 1933) states that $d_{TV}(\mu_t, \pi) \leq \Pr(X_t \neq Y_t)$.

To apply this, it is necessary to design a good coupling, so that quickly we are likely to have $X_t = Y_t$.

We must prove that this happens. The usual method is to use a metric $d(X_t, Y_t)$, and design the coupling to reduce it in expectation at each step.
Coupling

Coupling is one way of bounding the convergence time.

A coupling for a Markov chain \( \mathcal{M} \) is a random process \((X_t, Y_t)\) on \( \Omega \times \Omega \) such that each of \( X_t, Y_t \) is marginally a faithful copy of \( \mathcal{M} \).

If we assume that \((Y_t)\) starts with \( Y_0 \) in the equilibrium distribution, the Coupling Lemma (Doeblin, 1933) states that \( d_{TV}(p_t, \pi) \leq \Pr(X_t \neq Y_t) \).

To apply this, it is necessary to design a good coupling, so that quickly we are likely to have \( X_t = Y_t \).

We must prove that this happens. The usual method is to use a metric \( d(X_t, Y_t) \), and design the coupling to reduce it in expectation at each step.
Coupling

Coupling is one way of bounding the convergence time.

A coupling for a Markov chain $\mathcal{M}$ is a random process $(X_t, Y_t)$ on $\Omega \times \Omega$ such that each of $X_t$, $Y_t$ is marginally a faithful copy of $\mathcal{M}$.

If we assume that $(Y_t)$ starts with $Y_0$ in the equilibrium distribution, the Coupling Lemma (Doeblin, 1933) states that $d_{TV}(p_t, \pi) \leq \Pr(X_t \neq Y_t)$.

To apply this, it is necessary to design a good coupling, so that quickly we are likely to have $X_t = Y_t$.

We must prove that this happens. The usual method is to use a metric $d(X_t, Y_t)$, and design the coupling to reduce it in expectation at each step.
Consider again the independent sets example.

We can measure the distance $d(X, Y)$ between two independent sets $X, Y$ by the Hamming distance, i.e. the number of vertices at which they differ. This is a metric on $S$.

Now, to design a coupling, we have to specify the distribution of $(X_{t+1}, Y_{t+1})$ for all possible pairs of states $(X_t, Y_t)$. Then we have to prove that the distance is reduced in expectation at each step of the coupling. This may not be an easy task.

Path coupling reduces this difficulty by allowing the design of the coupling to be restricted to pairs such that $d(X_t, Y_t) = 1$. 
Consider again the independent sets example.

We can measure the distance \( d(X, Y) \) between two independent sets \( X, Y \) by the Hamming distance, i.e. the number of vertices at which they differ. This is a metric on \( S \).

Now, to design a coupling, we have to specify the distribution of \((X_{t+1}, Y_{t+1})\) for all possible pairs of states \((X_t, Y_t)\). Then we have to prove that the distance is reduced in expectation at each step of the coupling. This may not be an easy task.

Path coupling reduces this difficulty by allowing the design of the coupling to be restricted to pairs such that \( d(X_t, Y_t) = 1 \).
Consider again the independent sets example.

We can measure the distance \( d(X, Y) \) between two independent sets \( X, Y \) by the Hamming distance, i.e. the number of vertices at which they differ. This is a metric on \( S \).

Now, to design a coupling, we have to specify the distribution of \((X_{t+1}, Y_{t+1})\) for all possible pairs of states \((X_t, Y_t)\). Then we have to prove that the distance is reduced in expectation at each step of the coupling. This may not be an easy task.

Path coupling reduces this difficulty by allowing the design of the coupling to be restricted to pairs such that \( d(X_t, Y_t) = 1 \).
Consider again the independent sets example.

We can measure the distance \( d(X, Y) \) between two independent sets \( X, Y \) by the Hamming distance, i.e. the number of vertices at which they differ. This is a metric on \( S \).

Now, to design a coupling, we have to specify the distribution of \((X_{t+1}, Y_{t+1})\) for all possible pairs of states \((X_t, Y_t)\). Then we have to prove that the distance is reduced in expectation at each step of the coupling. This may not be an easy task.

Path coupling reduces this difficulty by allowing the design of the coupling to be restricted to pairs such that \( d(X_t, Y_t) = 1 \).
Path Coupling

Suppose $d(X_t, Y_t) = r$, and there exist states $Z_0, Z_1, \ldots, Z_r$ such that $Z_0 = X_t$, $Z_r = Y_t$ and $d(Z_{i-1}, Z_i) = 1$ ($i = 1, 2, \ldots, r$).

In the independent sets example this can be realised by first deleting $X_t \setminus Y_t$, and then inserting $Y_t \setminus X_t$, one vertex at a time.

Now let all $Z_i$ follow the Markov chain (according to some coupling) for one step to give the evolved states $Z'_0, Z'_1, \ldots, Z'_r$.

So $X_{t+1} = Z'_0$, $Y_{t+1} = Z'_r$.

Suppose we can show that $\mathbb{E}[d(Z'_{i-1}, Z'_i)] \leq 1 - \beta$ ($i = 1, 2, \ldots, r$), for some $\beta > 0$. 
Suppose $d(X_t, Y_t) = r$, and there exist states $Z_0, Z_1, \ldots, Z_r$ such that $Z_0 = X_t$, $Z_r = Y_t$ and $d(Z_{i-1}, Z_i) = 1$ ($i = 1, 2, \ldots, r$).

In the independent sets example this can be realised by first deleting $X_t \setminus Y_t$, and then inserting $Y_t \setminus X_t$, one vertex at a time.

Now let all $Z_i$ follow the Markov chain (according to some coupling) for one step to give the evolved states $Z'_0, Z'_1, \ldots, Z'_r$.

So $X_{t+1} = Z'_0$, $Y_{t+1} = Z'_r$.

Suppose we can show that $\mathbb{E}[d(Z'_{i-1}, Z'_i)] \leq 1 - \beta$ ($i = 1, 2, \ldots, r$), for some $\beta > 0$. 
Path Coupling

Suppose \( d(X_t, Y_t) = r \), and there exist states \( Z_0, Z_1, \ldots, Z_r \) such that \( Z_0 = X_t, Z_r = Y_t \) and \( d(Z_{i-1}, Z_i) = 1 \) \( (i = 1, 2, \ldots, r) \).

In the independent sets example this can be realised by first deleting \( X_t \setminus Y_t \), and then inserting \( Y_t \setminus X_t \), one vertex at a time.

Now let all \( Z_i \) follow the Markov chain (according to some coupling) for one step to give the evolved states \( Z'_0, Z'_1, \ldots, Z'_r \).

So \( X_{t+1} = Z'_0, Y_{t+1} = Z'_r \).

Suppose we can show that \( \mathbb{E}[d(Z'_{i-1}, Z'_i)] \leq 1 - \beta \) \( (i = 1, 2, \ldots, r) \), for some \( \beta > 0 \).
Path Coupling

Suppose $d(X_t, Y_t) = r$, and there exist states $Z_0, Z_1, \ldots, Z_r$ such that $Z_0 = X_t$, $Z_r = Y_t$ and $d(Z_{i-1}, Z_i) = 1$ ($i = 1, 2, \ldots, r$).

In the independent sets example this can be realised by first deleting $X_t \setminus Y_t$, and then inserting $Y_t \setminus X_t$, one vertex at a time.

Now let all $Z_i$ follow the Markov chain (according to some coupling) for one step to give the evolved states $Z'_0, Z'_1, \ldots, Z'_r$.

So $X_{t+1} = Z'_0$, $Y_{t+1} = Z'_r$.

Suppose we can show that $\mathbb{E}[d(Z'_{i-1}, Z'_i)] \leq 1 - \beta$ ($i = 1, 2, \ldots, r$), for some $\beta > 0$. 
Suppose $d(X_t, Y_t) = r$, and there exist states $Z_0, Z_1, \ldots, Z_r$ such that $Z_0 = X_t$, $Z_r = Y_t$ and $d(Z_{i-1}, Z_i) = 1$ ($i = 1, 2, \ldots, r$).

In the independent sets example this can be realised by first deleting $X_t \setminus Y_t$, and then inserting $Y_t \setminus X_t$, one vertex at a time.

Now let all $Z_i$ follow the Markov chain (according to some coupling) for one step to give the evolved states $Z'_0, Z'_1, \ldots, Z'_r$.

So $X_{t+1} = Z'_0$, $Y_{t+1} = Z'_r$.

Suppose we can show that $\mathbb{E}[d(Z'_{i-1}, Z'_i)] \leq 1 - \beta$ ($i = 1, 2, \ldots, r$), for some $\beta > 0$. 
Then, following Bubley and Dyer, 1997, we have

\[ \mathbb{E}[d(X_{t+1}, Y_{t+1})] \leq \mathbb{E}[\sum_{i=1}^r d(Z'_{i-1}, Z'_i)] \quad \text{(metric property)} \]

\[ = \sum_{i=1}^r \mathbb{E}[d(Z'_{i-1}, Z'_i)] \quad \text{(linearity of expectation)} \]

\[ \leq (1 - \beta)r \quad \text{(by assumption)} \]

\[ = (1 - \beta) d(X_t, Y_t). \]

So \( d_{TV}(p_t, \pi) \leq \Pr(X_t \neq Y_t) \quad \text{(coupling lemma)} \)

\[ \leq \mathbb{E}[d(X_t, Y_t)] \quad \text{(first moment)} \]

\[ \leq (1 - \beta)^t \mathbb{E}[d(X_0, Y_0)] \quad \text{(from above)} \]

\[ \leq n(1 - \beta)^t \leq ne^{-\beta t}. \]

So, for \( t \geq (\ln n + c)/\beta \), we will have \( d_{TV}(p_t, \pi) \leq e^{-c} \).
Then, following Bubley and Dyer, 1997, we have

\[
\mathbb{E}[d(X_{t+1}, Y_{t+1})] \leq \mathbb{E}\left[\sum_{i=1}^{r} d(Z'_{i-1}, Z'_i)\right] \quad \text{(metric property)}
\]

\[
= \sum_{i=1}^{r} \mathbb{E}[d(Z'_{i-1}, Z'_i)] \quad \text{(linearity of expectation)}
\]

\[
\leq (1 - \beta)r \quad \text{(by assumption)}
\]

\[
= (1 - \beta)d(X_t, Y_t).
\]

So \(d_{TV}(p_t, \pi) \leq \Pr(X_t \neq Y_t) \quad \text{(coupling lemma)}\)

\[
\leq \mathbb{E}[d(X_t, Y_t)] \quad \text{(first moment)}
\]

\[
\leq (1 - \beta)^t \mathbb{E}[d(X_0, Y_0)] \quad \text{(from above)}
\]

\[
\leq n(1 - \beta)^t \leq ne^{-\beta t}.
\]

So, for \(t \geq (\ln n + c)/\beta\), we will have \(d_{TV}(p_t, \pi) \leq e^{-c}\).
Then, following Bubley and Dyer, 1997, we have

\[ \mathbb{E}[d(X_{t+1}, Y_{t+1})] \leq \mathbb{E}\left[ \sum_{i=1}^{r} d(Z'_{i-1}, Z'_i) \right] \quad \text{(metric property)} \]

\[ = \sum_{i=1}^{r} \mathbb{E}\left[ d(Z'_{i-1}, Z'_i) \right] \quad \text{(linearity of expectation)} \]

\[ \leq (1 - \beta)r \quad \text{(by assumption)} \]

\[ = (1 - \beta)d(X_t, Y_t). \]

So \( d_{TV}(p_t, \pi) \leq \Pr(X_t \neq Y_t) \quad \text{(coupling lemma)} \)

\[ \leq \mathbb{E}[d(X_t, Y_t)] \quad \text{(first moment)} \]

\[ \leq (1 - \beta)^t \mathbb{E}[d(X_0, Y_0)] \quad \text{(from above)} \]

\[ \leq n(1 - \beta)^t \leq ne^{-\beta t}. \]

So, for \( t \geq (\ln n + c)/\beta \), we will have \( d_{TV}(p_t, \pi) \leq e^{-c}. \)
Then, following Bubley and Dyer, 1997, we have

\[
\mathbb{E}[d(X_{t+1}, Y_{t+1})] \leq \mathbb{E}\left[\sum_{i=1}^{r} d(Z'_{i-1}, Z'_i)\right] \quad \text{(metric property)}
\]

\[
= \sum_{i=1}^{r} \mathbb{E}[d(Z'_{i-1}, Z'_i)] \quad \text{(linearity of expectation)}
\]

\[
\leq (1 - \beta)r \quad \text{(by assumption)}
\]

\[
= (1 - \beta)d(X_t, Y_t).
\]

So \(d_{TV}(p_t, \pi) \leq \Pr(X_t \neq Y_t) \quad \text{(coupling lemma)}\)

\[
\leq \mathbb{E}[d(X_t, Y_t)] \quad \text{(first moment)}
\]

\[
\leq (1 - \beta)^t \mathbb{E}[d(X_0, Y_0)] \quad \text{(from above)}
\]

\[
\leq n(1 - \beta)^t \leq ne^{-\beta t}.
\]

So, for \(t \geq (\ln n + c)/\beta\), we will have \(d_{TV}(p_t, \pi) \leq e^{-c}\).
Then, following Bubley and Dyer, 1997, we have

\[
\mathbb{E}[d(X_{t+1}, Y_{t+1})] \leq \mathbb{E}\left[ \sum_{i=1}^{r} d(Z'_{i-1}, Z'_i) \right] \quad \text{(metric property)}
\]

\[
= \sum_{i=1}^{r} \mathbb{E}[d(Z'_{i-1}, Z'_i)] \quad \text{(linearity of expectation)}
\]

\[
\leq (1 - \beta) r \quad \text{(by assumption)}
\]

\[
= (1 - \beta)d(X_t, Y_t).
\]

So \(d_{TV}(\rho_t, \pi) \leq \Pr(X_t \neq Y_t) \quad \text{(coupling lemma)}\)

\[
\leq \mathbb{E}[d(X_t, Y_t)] \quad \text{(first moment)}
\]

\[
\leq (1 - \beta)^t \mathbb{E}[d(X_0, Y_0)] \quad \text{(from above)}
\]

\[
\leq n(1 - \beta)^t \leq ne^{-\beta t}.
\]

So, for \(t \geq (\ln n + c)/\beta\), we will have \(d_{TV}(\rho_t, \pi) \leq e^{-c}\).
Then, following Bubley and Dyer, 1997, we have

\[
\mathbb{E}[d(X_{t+1}, Y_{t+1})] \leq \mathbb{E}\left[\sum_{i=1}^{r} d(Z'_{i-1}, Z'_i)\right] \quad \text{(metric property)}
\]

\[
= \sum_{i=1}^{r} \mathbb{E}[d(Z'_{i-1}, Z'_i)] \quad \text{(linearity of expectation)}
\]

\[
\leq (1 - \beta) r \quad \text{(by assumption)}
\]

\[
= (1 - \beta) d(X_t, Y_t).
\]

So \(d_{TV}(p_t, \pi) \leq \Pr(X_t \neq Y_t) \quad \text{(coupling lemma)}\)

\[
\leq \mathbb{E}[d(X_t, Y_t)] \quad \text{(first moment)}
\]

\[
\leq (1 - \beta)^t \mathbb{E}[d(X_0, Y_0)] \quad \text{(from above)}
\]

\[
\leq n(1 - \beta)^t \leq ne^{-\beta t}.
\]

So, for \(t \geq (\ln n + c)/\beta\), we will have \(d_{TV}(p_t, \pi) \leq e^{-c}\).
Path Coupling

Then, following Bubley and Dyer, 1997, we have

$$
\mathbb{E}[d(X_{t+1}, Y_{t+1})] \leq \mathbb{E}\left[ \sum_{i=1}^{r} d(Z'_{i-1}, Z'_i) \right] \quad \text{(metric property)}
$$

$$
= \sum_{i=1}^{r} \mathbb{E}[d(Z'_{i-1}, Z'_i)] \quad \text{(linearity of expectation)}
$$

$$
\leq (1 - \beta)r \quad \text{(by assumption)}
$$

$$
= (1 - \beta)d(X_t, Y_t).
$$

So $d_{TV}(p_t, \pi) \leq \Pr(X_t \neq Y_t) \quad \text{(coupling lemma)}$

$$
\leq \mathbb{E}[d(X_t, Y_t)] \quad \text{(first moment)}
$$

$$
\leq (1 - \beta)t \mathbb{E}[d(X_0, Y_0)] \quad \text{(from above)}
$$

$$
\leq n(1 - \beta)^t \leq ne^{-\beta t}.
$$

So, for $t \geq (\ln n + c)/\beta$, we will have $d_{TV}(p_t, \pi) \leq e^{-c}$. 
Then, following Bubley and Dyer, 1997, we have

\[
\mathbb{E}[d(X_{t+1}, Y_{t+1})] \leq \mathbb{E}[\sum_{i=1}^{r} d(Z'_{i-1}, Z'_i)] \quad \text{(metric property)}
\]

\[
= \sum_{i=1}^{r} \mathbb{E}[d(Z'_{i-1}, Z'_i)] \quad \text{(linearity of expectation)}
\]

\[
\leq (1 - \beta)r \quad \text{(by assumption)}
\]

\[
= (1 - \beta)d(X_t, Y_t).
\]

So \(d_{TV}(p_t, \pi) \leq \Pr(X_t \neq Y_t) \quad \text{(coupling lemma)}
\]

\[
\leq \mathbb{E}[d(X_t, Y_t)] \quad \text{(first moment)}
\]

\[
\leq (1 - \beta)^t \mathbb{E}[d(X_0, Y_0)] \quad \text{(from above)}
\]

\[
\leq n(1 - \beta)^t \leq ne^{-\beta t}.
\]

So, for \(t \geq (\ln n + c)/\beta\), we will have \(d_{TV}(p_t, \pi) \leq e^{-c}\).
Then, following Bubley and Dyer, 1997, we have

\[
\mathbb{E}[d(X_{t+1}, Y_{t+1})] \leq \mathbb{E}\left[\sum_{i=1}^{r} d(Z'_{i-1}, Z'_i)\right] \quad \text{(metric property)}
\]

\[
= \sum_{i=1}^{r} \mathbb{E}\left[d(Z'_{i-1}, Z'_i)\right] \quad \text{(linearity of expectation)}
\]

\[
\leq (1 - \beta)r \quad \text{(by assumption)}
\]

\[
= (1 - \beta)d(X_t, Y_t).
\]

So \(d_{TV}(p_t, \pi) \leq \Pr(X_t \neq Y_t)\) \quad \text{(coupling lemma)}

\[
\leq \mathbb{E}[d(X_t, Y_t)] \quad \text{(first moment)}
\]

\[
\leq (1 - \beta)^t \mathbb{E}[d(X_0, Y_0)] \quad \text{(from above)}
\]

\[
\leq n(1 - \beta)^t \leq ne^{-\beta t}.
\]

So, for \(t \geq (\ln n + c)/\beta\), we will have \(d_{TV}(p_t, \pi) \leq e^{-c}\).
Example: Independent sets

Consider the following coupling:

If \( v \) chosen, delete in \( X \), do nothing in \( Y \), with probability \( \frac{1}{2} \),
do nothing in \( X \), insert in \( Y \), with probability \( \frac{1}{2} \).

If any other vertex is chosen, attempt the same move in both
\( X \) and \( Y \). The outcome differs only if a neighbour of \( v \) is chosen.
Consider the following coupling:

If $v$ chosen, delete in $X$, do nothing in $Y$, with probability $\frac{1}{2}$,
  do nothing in $X$, insert in $Y$, with probability $\frac{1}{2}$.

If any other vertex is chosen, attempt the same move in both $X$ and $Y$. The outcome differs only if a neighbour of $v$ is chosen.
Example: Independent sets

Vertex $v \in X$, $v \notin Y$.

Consider the following coupling:

If $v$ chosen, delete in $X$, do nothing in $Y$, with probability $\frac{1}{2}$,
  do nothing in $X$, insert in $Y$, with probability $\frac{1}{2}$.

If any other vertex is chosen, attempt the same move in both $X$ and $Y$. The outcome differs only if a neighbour of $v$ is chosen.
Analysis of the coupling

A bad configuration for our coupling is when all neighbours of $v$ have no neighbours in $X$ and $Y$ except $v$.

Recall the metric $d$ is the Hamming distance between $X$ and $Y$.

Thus, since $G$ has $n$ vertices and maximum degree $\Delta$, we require

$$\mathbb{E}[d(X_{t+1}, Y_{t+1})] \leq 1 - \frac{1}{n} + \frac{\Delta}{2n} \leq 1.$$  

This is only true if $\Delta \leq 2$, a rather weak result.

A better result, $\Delta \leq 4$, can be obtained, by using either a better Markov chain or a better metric.

(Luby and Vigoda, 1999, Dyer and Greenhill, 2000.)
Analysis of the coupling

A bad configuration for our coupling is when all neighbours of $v$ have no neighbours in $X$ and $Y$ except $v$.

Recall the metric $d$ is the Hamming distance between $X$ and $Y$.

Thus, since $G$ has $n$ vertices and maximum degree $\Delta$, we require

$$\mathbb{E}[d(X_{t+1}, Y_{t+1})] \leq 1 - \frac{1}{n} + \frac{\Delta}{2n} \leq 1.$$  

This is only true if $\Delta \leq 2$, a rather weak result.

A better result, $\Delta \leq 4$, can be obtained, by using either a better Markov chain or a better metric.

(Luby and Vigoda, 1999, Dyer and Greenhill, 2000.)
Analysis of the coupling

A bad configuration for our coupling is when all neighbours of $v$ have no neighbours in $X$ and $Y$ except $v$.

Recall the metric $d$ is the Hamming distance between $X$ and $Y$.

Thus, since $G$ has $n$ vertices and maximum degree $\Delta$, we require

$$\mathbb{E}[d(X_{t+1}, Y_{t+1})] \leq 1 - \frac{1}{n} + \frac{\Delta}{2n} \leq 1.$$ 

This is only true if $\Delta \leq 2$, a rather weak result.

A better result, $\Delta \leq 4$, can be obtained, by using either a better Markov chain or a better metric.

(Luby and Vigoda, 1999, Dyer and Greenhill, 2000.)
A bad configuration for our coupling is when all neighbours of $v$ have no neighbours in $X$ and $Y$ except $v$.

Recall the metric $d$ is the Hamming distance between $X$ and $Y$.

Thus, since $G$ has $n$ vertices and maximum degree $\Delta$, we require

$$\mathbb{E}[d(X_{t+1}, Y_{t+1})] \leq 1 - \frac{1}{n} + \frac{\Delta}{2n} \leq 1.$$ 

This is only true if $\Delta \leq 2$, a rather weak result.

A better result, $\Delta \leq 4$, can be obtained, by using either a better Markov chain or a better metric.

(Luby and Vigoda, 1999, Dyer and Greenhill, 2000.)
Analysis of the coupling

A bad configuration for our coupling is when all neighbours of $v$ have no neighbours in $X$ and $Y$ except $v$.

Recall the metric $d$ is the Hamming distance between $X$ and $Y$.

Thus, since $G$ has $n$ vertices and maximum degree $\Delta$, we require

$$\mathbb{E}[d(X_{t+1}, Y_{t+1})] \leq 1 - \frac{1}{n} + \frac{\Delta}{2n} \leq 1.$$ 

This is only true if $\Delta \leq 2$, a rather weak result.

A better result, $\Delta \leq 4$, can be obtained, by using either a better Markov chain or a better metric.

(Luby and Vigoda, 1999, Dyer and Greenhill, 2000.)
Analysis of the coupling

A bad configuration for our coupling is when all neighbours of $v$ have no neighbours in $X$ and $Y$ except $v$.

Recall the metric $d$ is the Hamming distance between $X$ and $Y$.

Thus, since $G$ has $n$ vertices and maximum degree $\Delta$, we require

$$\mathbb{E}[d(X_{t+1}, Y_{t+1})] \leq 1 - \frac{1}{n} + \frac{\Delta}{2n} \leq 1.$$  

This is only true if $\Delta \leq 2$, a rather weak result.

A better result, $\Delta \leq 4$, can be obtained, by using either a better Markov chain or a better metric.

(Luby and Vigoda, 1999, Dyer and Greenhill, 2000.)
Developments of coupling

The main problem with coupling is that we always have to consider the worst case configuration for $X$, $Y$. This gives pessimistic results, which can often be very weak. Sometimes the results can be improved by judicious choice of chain and/or metric, but this increases the difficulty of analysis.

Two ideas have been used to try to deal with this problem.

- Burn-in
- Stopping times

We will briefly describe these, and give an example of the stopping time approach.
Developments of coupling

The main problem with coupling is that we always have to consider the worst case configuration for $X$, $Y$. This gives pessimistic results, which can often be very weak. Sometimes the results can be improved by judicious choice of chain and/or metric, but this increases the difficulty of analysis.

Two ideas have been used to try to deal with this problem.

- **Burn-in**
- **Stopping times**

We will briefly describe these, and give an example of the stopping time approach.
Developments of coupling

The main problem with coupling is that we always have to consider the worst case configuration for $X, Y$. This gives pessimistic results, which can often be very weak. Sometimes the results can be improved by judicious choice of chain and/or metric, but this increases the difficulty of analysis.

Two ideas have been used to try to deal with this problem.

- **Burn-in**
- **Stopping times**

We will briefly describe these, and give an example of the stopping time approach.
Developments of coupling

The main problem with coupling is that we always have to consider the worst case configuration for $X$, $Y$. This gives pessimistic results, which can often be very weak. Sometimes the results can be improved by judicious choice of chain and/or metric, but this increases the difficulty of analysis.

Two ideas have been used to try to deal with this problem.

- Burn-in
- Stopping times

We will briefly describe these, and give an example of the stopping time approach.
Developments of coupling

The main problem with coupling is that we always have to consider the worst case configuration for $X$, $Y$. This gives pessimistic results, which can often be very weak. Sometimes the results can be improved by judicious choice of chain and/or metric, but this increases the difficulty of analysis.

Two ideas have been used to try to deal with this problem.

- Burn-in
- Stopping times

We will briefly describe these, and give an example of the stopping time approach.
Developments of coupling

- **Burn-in**
  Dyer and Frieze, 2001, used the idea that $X_t$ may quickly acquire some of the properties of a random state.
  The approach is to consider a relatively short burn-in period and to show that, after this, $X_t$ has useful properties which are not shared by the worst care configurations.

- **Stopping times**
  Dyer, Goldberg, Greenhill, Jerrum and Mitzenmacher, 2000, used the idea of running path coupling on a pair $X$, $Y$ for multiple steps, until the distance changes.
  When this stopping time occurs, with reasonable probability we may no longer be in a worst case configuration.
Developments of coupling

- **Burn-in**
  Dyer and Frieze, 2001, used the idea that $X_t$ may quickly acquire some of the properties of a random state. The approach is to consider a relatively short burn-in period and to show that, after this, $X_t$ has useful properties which are not shared by the worst care configurations.

- **Stopping times**
  Dyer, Goldberg, Greenhill, Jerrum and Mitzenmacher, 2000, used the idea of running path coupling on a pair $X, Y$ for multiple steps, until the distance changes. When this stopping time occurs, with reasonable probability we may no longer be in a worst case configuration.
Developments of coupling

- **Burn-in**
  
  Dyer and Frieze, 2001, used the idea that $X_t$ may quickly acquire some of the properties of a random state. The approach is to consider a relatively short burn-in period and to show that, after this, $X_t$ has useful properties which are not shared by the worst care configurations.

- **Stopping times**
  
  Dyer, Goldberg, Greenhill, Jerrum and Mitzenmacher, 2000, used the idea of running path coupling on a pair $X$, $Y$ for multiple steps, until the distance changes. When this stopping time occurs, with reasonable probability we may no longer be in a worst case configuration.
Developments of coupling

- **Burn-in**
  Dyer and Frieze, 2001, used the idea that $X_t$ may quickly acquire some of the properties of a random state. The approach is to consider a relatively short burn-in period and to show that, after this, $X_t$ has useful properties which are not shared by the worst care configurations.

- **Stopping times**
  Dyer, Goldberg, Greenhill, Jerrum and Mitzenmacher, 2000, used the idea of running path coupling on a pair $X, Y$ for multiple steps, until the distance changes. When this stopping time occurs, with reasonable probability we may no longer be in a worst case configuration.
Stopping times were first used by Dyer, Greenhill, Goldberg, Jerrum and Mitzenmacher, 2001, for a Markov chain on graph colourings. The idea was used to obtain a small improvement on a result of Jerrum, 1995, that rapid mixing occurs with $q$ colours if $q \geq 2\Delta$.

Stopping times were first used by Dyer, Greenhill, Goldberg, Jerrum and Mitzenmacher, 2001, for a Markov chain on graph colourings. The idea was used to obtain a small improvement on a result of Jerrum, 1995, that rapid mixing occurs with $q$ colours if $q \geq 2\Delta$.

Example: Hypergraph independent sets

Set system: $\mathcal{H} = (V, E)$. Vertices $v$ are the elements of $V$, where $|V| = n$. Edge set $E$ contains subsets $e$ of $V$. Degree of $v \in V$ is the number of edges containing $v$.

An independent set is any subset $I$ of $V$ such that no edge of $\mathcal{H}$ is contained entirely in $I$. Generalises the graph case, where all edges are sets of size 2.

We consider hypergraphs with minimum edge size $m > 2$ and maximum degree $\Delta$. 
Example: Hypergraph independent sets

Set system: $\mathcal{H} = (V, E)$. Vertices $v$ are the elements of $V$, where $|V| = n$. Edge set $E$ contains subsets $e$ of $V$. Degree of $v \in V$ is the number of edges containing $v$.

An independent set is any subset $I$ of $V$ such that no edge of $\mathcal{H}$ is contained entirely in $I$. Generalises the graph case, where all edges are sets of size 2.

We consider hypergraphs with minimum edge size $m > 2$ and maximum degree $\Delta$. 
Example: Hypergraph independent sets

Set system: $\mathcal{H} = (V, E)$. Vertices $v$ are the elements of $V$, where $|V| = n$. Edge set $E$ contains subsets $e$ of $V$. Degree of $v \in V$ is the number of edges containing $v$.

An independent set is any subset $I$ of $V$ such that no edge of $\mathcal{H}$ is contained entirely in $I$. Generalises the graph case, where all edges are sets of size 2.

We consider hypergraphs with minimum edge size $m > 2$ and maximum degree $\Delta$. 
The Markov chain for graphs can also be used for hypergraphs with the obvious modifications.

Unfortunately, exactly the same analysis applies here also.

Bad configuration: Change vertex \( v \) surrounded by almost full edges.

Therefore we cannot directly prove convergence of this chain using path coupling if \( \Delta > 2 \).
The Markov chain for graphs can also be used for hypergraphs with the obvious modifications.

Unfortunately, exactly the same analysis applies here also.

Bad configuration: Change vertex $v$ surrounded by almost full edges.

Therefore we cannot directly prove convergence of this chain using path coupling if $\Delta > 2$. 
Example: Hypergraph independent sets (2)

The Markov chain for graphs can also be used for hypergraphs with the obvious modifications.

Unfortunately, exactly the same analysis applies here also.

Bad configuration: Change vertex $v$ surrounded by almost full edges.

Therefore we cannot directly prove convergence of this chain using path coupling if $\Delta > 2$. 
Consider a bad configuration. In one step it is more likely that a bad vertex like $w$ (increasing Hamming distance) is chosen than the single good vertex $v$ (decreasing Hamming distance).

But it is even more likely that a vertex in an edge $e$ containing $v$ is removed from $I$. If $e$ has two vertices other than $v$ not in $I$, any vertex in $e$ can be added in both $X$ and $Y$. So, if $T$ is defined as the time at which the distance between $X_t$ and $Y_t$ first changes, the expected distance between $X_T$ and $Y_T$ may be less than 1.
Intuition

Consider a bad configuration. In one step it is more likely that a bad vertex like $w$ (increasing Hamming distance) is chosen than the single good vertex $v$ (decreasing Hamming distance).

But it is even more likely that a vertex in an edge $e$ containing $v$ is removed from $I$. If $e$ has two vertices other than $v$ not in $I$, any vertex in $e$ can be added in both $X$ and $Y$. So, if $T$ is defined as the time at which the distance between $X_t$ and $Y_t$ first changes, the expected distance between $X_T$ and $Y_T$ may be less than 1.
Theorem [Bordewich, Dyer and Karpinski, 2005]
Let $\mathcal{M}$ be a Markov chain on state space $\Omega$. Let $(X_t, Y_t)$ be a path coupling for $\mathcal{M}$ and let $d$ be an integer valued metric on $\Omega \times \Omega$.
For any states $X_0$ and $Y_0$ such that $d(X_0, Y_0) = 1$, define $T = \min\{t : d(X_t, Y_t) \neq 1\}$. Suppose, for some $p > 0$, that

(a) $\Pr(T = t \mid T \geq t) \geq p$, independently for each $t$,
(b) $\mathbb{E}[d(X_T, Y_T)] \leq \alpha < 1$.

Then the mixing time of $\mathcal{M}$ satisfies

$$\tau(\epsilon) \leq \frac{1}{p} \frac{3}{1 - \alpha} \ln(eD_2) \ln\left(\frac{2D_1}{\epsilon(1 - \alpha)}\right),$$

where $D_1 = \max\{d(X, Y) : X, Y \in \Omega\}$ and $D_2 = \max\{d(X_T, Y_T) : X_0, Y_0 \in \Omega, d(X_0, Y_0) = 1\}$. 
To use this, we must analyse what happens in a bad configuration. We simplify by analysing this for a single edge $e$ in the configuration. This does not give the best possible result, but makes the analysis tractable.

Process:
At each step, pick a vertex $u$ at random. If $u = v$, we stop with success.
If $u \in I$, remove it with probability $\frac{1}{2}$.
If $u \notin I$, insert it with probability $\frac{1}{2}$.
If all vertices are occupied, we stop with failure, otherwise we proceed to the next step.
To use this, we must analyse what happens in a bad configuration. We simplify by analysing this for a single edge $e$ in the configuration. This does not give the best possible result, but makes the analysis tractable.

**Process:**
At each step, pick a vertex $u$ at random. If $u = v$, we stop with success.
If $u \in I$, remove it with probability $\frac{1}{2}$.
If $u \notin I$, insert it with probability $\frac{1}{2}$.
If all vertices are occupied, we stop with failure, otherwise we proceed to the next step.
Analysis of edge process

Let $p_k$ denote the probability that we stop with failure, given that we start with $k$ vertices in the edge not in $I$. We obtain the following system of simultaneous equations.

\[
(m + 1)p_1 - (m - 2)p_2 = 1
\]
\[
-kp_{k-1} + (m + 1)p_k - (m - k - 1)p_{k+1} = 0
\]

$k = 2, 3, \ldots, m - 1$

We can show that these have solution

\[
p_k = \frac{1}{\binom{m-1}{k}} \left( 1 - \frac{\sum_{i=1}^{k} \binom{m}{i}}{2^m - 1} \right) \ (k = 1, 2, \ldots, m - 1).\]

In particular, $p_k$ decreases with $k$, and

\[
p_1 = \frac{1}{m - 1} \left( 1 - \frac{m}{2^m - 1} \right).\]
Analysis of edge process

Let $p_k$ denote the probability that we stop with failure, given that we start with $k$ vertices in the edge not in $I$. We obtain the following system of simultaneous equations.

\[
(m + 1)p_1 - (m - 2)p_2 = 1 \\
-kp_{k-1} + (m + 1)p_k - (m - k - 1)p_{k+1} = 0 \\
\]

$(k = 2, 3, \ldots, m - 1)$

We can show that these have solution

\[
p_k = \frac{1}{(m-1)^k} \left( 1 - \frac{\sum_{i=1}^{k} \binom{m}{i}}{2^m - 1} \right) \quad (k = 1, 2, \ldots, m - 1). \]

In particular, $p_k$ decreases with $k$, and

\[
p_1 = \frac{1}{m-1} \left( 1 - \frac{m}{2^m - 1} \right). \]
Analysis of edge process

Let $p_k$ denote the probability that we stop with failure, given that we start with $k$ vertices in the edge not in $I$. We obtain the following system of simultaneous equations.

\[
(m + 1)p_1 - (m - 2)p_2 = 1 \\
-kp_{k-1} + (m + 1)p_k - (m - k - 1)p_{k+1} = 0
\]

\( (k = 2, 3, \ldots, m - 1) \)

We can show that these have solution

\[
p_k = \frac{1}{\binom{m-1}{k}} \left( 1 - \sum_{i=1}^{k} \frac{\binom{m}{i}}{2^m - 1} \right) \quad (k = 1, 2, \ldots, m - 1).
\]

In particular, $p_k$ decreases with $k$, and

\[
p_1 = \frac{1}{m-1} \left( 1 - \frac{m}{2^m - 1} \right).
\]
Analysis of coupling

Since $v$ is in at most $\Delta$ edges, the probability that we have a failure on any edge is at most $\Delta p_1$. The probability that the stopping time ends with success is therefore at least $1 - \Delta p_1$.

Now success means the Hamming distance decreases by 1, and failure means the Hamming distance increases by 1. Thus we have

$$\mathbb{E}[d(X_T, Y_T)] \leq 1 - (1 - \Delta p_1) + \Delta p_1 = 2\Delta p_1 < 1,$$

provided

$$2\Delta p_1 = \frac{2\Delta}{m-1} \left(1 - \frac{m}{2^m - 1} \right) < 1.$$
Analysis of coupling

Since $v$ is in at most $\Delta$ edges, the probability that we have a failure on any edge is at most $\Delta p_1$. The probability that the stopping time ends with success is therefore at least $1 - \Delta p_1$.

Now success means the Hamming distance decreases by 1, and failure means the Hamming distance increases by 1. Thus we have

$$\mathbb{E}[d(X_T, Y_T)] \leq 1 - (1 - \Delta p_1) + \Delta p_1 = 2\Delta p_1 < 1,$$

provided

$$2\Delta p_1 = \frac{2\Delta}{m-1} \left(1 - \frac{m}{2^m - 1}\right) < 1.$$
Analysis of coupling

Since \( v \) is in at most \( \Delta \) edges, the probability that we have a failure on any edge is at most \( \Delta p_1 \). The probability that the stopping time ends with success is therefore at least \( 1 - \Delta p_1 \).

Now success means the Hamming distance decreases by 1, and failure means the Hamming distance increases by 1. Thus we have

\[
\mathbb{E}[d(X_T, Y_T)] \leq 1 - (1 - \Delta p_1) + \Delta p_1 = 2\Delta p_1 < 1,
\]

provided

\[
2\Delta p_1 = \frac{2\Delta}{m-1} \left( 1 - \frac{m}{2^m - 1} \right) < 1.
\]
It is clearly sufficient that

\[
\frac{2\Delta}{m - 1} \leq 1,
\]

i.e. \( m \geq 2\Delta + 1 \).

Thus we have polynomial time approximate counting for independent sets in hypergraphs of maximum degree \( \Delta \) provided the minimum edge size \( m \) satisfies \( m \geq 2\Delta + 1 \).

The only lower bound we have is that approximate counting is provably impossible for \( m = o(\log \Delta) \), unless \( \text{NP} = \text{RP} \).
It is clearly sufficient that
\[ \frac{2\Delta}{m - 1} \leq 1, \]
i.e. \( m \geq 2\Delta + 1 \).

Thus we have polynomial time approximate counting for independent sets in hypergraphs of maximum degree \( \Delta \) provided the minimum edge size \( m \) satisfies \( m \geq 2\Delta + 1 \).

The only lower bound we have is that approximate counting is provably impossible for \( m = o(\log \Delta) \), unless \( NP = RP \).
Approximate counting

It is clearly sufficient that

\[ \frac{2\Delta}{m - 1} \leq 1, \]

i.e. \( m \geq 2\Delta + 1. \)

Thus we have polynomial time approximate counting for independent sets in hypergraphs of maximum degree \( \Delta \) provided the minimum edge size \( m \) satisfies \( m \geq 2\Delta + 1. \)

The only lower bound we have is that approximate counting is provably impossible for \( m = o(\log \Delta) \), unless \( NP = RP \).
Non-Markovian coupling

The couplings we have described are Markovian, meaning that the joint process \((X_t, Y_t)\) is itself a Markov chain.

It is known that

- For any Markov chain, there is a coupling which exactly attains the mixing time of the chain. (Griffeath, 1975)
- This coupling is not in general Markovian. (Burdzy and Kendall, 1998, Anil Kumar and Ramesh, 2001)

Non-Markovian couplings are very difficult to analyse. Hayes and Vigoda, 2003, analysed a coupling for graph colourings with some non-Markovian evolution, but there have been few other attempts.

Other techniques are necessary where coupling seems inapplicable.
Non-Markovian coupling

The couplings we have described are Markovian, meaning that the joint process \((X_t, Y_t)\) is itself a Markov chain.

It is known that

- For any Markov chain, there is a coupling which exactly attains the mixing time of the chain. (Griffeath, 1975)
- This coupling is not in general Markovian. (Burdzy and Kendall, 1998, Anil Kumar and Ramesh, 2001)

Non-Markovian couplings are very difficult to analyse. Hayes and Vigoda, 2003, analysed a coupling for graph colourings with some non-Markovian evolution, but there have been few other attempts.

Other techniques are necessary where coupling seems inapplicable.
Non-Markovian coupling

The couplings we have described are Markovian, meaning that the joint process \((X_t, Y_t)\) is itself a Markov chain.

It is known that

- For any Markov chain, there is a coupling which exactly attains the mixing time of the chain. (Griffeath, 1975)
- This coupling is not in general Markovian. (Burdzy and Kendall, 1998, Anil Kumar and Ramesh, 2001)

Non-Markovian couplings are very difficult to analyse. Hayes and Vigoda, 2003, analysed a coupling for graph colourings with some non-Markovian evolution, but there have been few other attempts.

Other techniques are necessary where coupling seems inapplicable.
The couplings we have described are Markovian, meaning that the joint process \((X_t, Y_t)\) is itself a Markov chain.

It is known that

- For any Markov chain, there is a coupling which exactly attains the mixing time of the chain. (Griffeath, 1975)
- This coupling is not in general Markovian. (Burdzy and Kendall, 1998, Anil Kumar and Ramesh, 2001)

Non-Markovian couplings are very difficult to analyse. Hayes and Vigoda, 2003, analysed a coupling for graph colourings with some non-Markovian evolution, but there have been few other attempts.

Other techniques are necessary where coupling seems inapplicable.
Non-Markovian coupling

The couplings we have described are Markovian, meaning that the joint process \((X_t, Y_t)\) is itself a Markov chain.

It is known that

- For any Markov chain, there is a coupling which exactly attains the mixing time of the chain. (Griffeath, 1975)
- This coupling is not in general Markovian. (Burdzy and Kendall, 1998, Anil Kumar and Ramesh, 2001)

Non-Markovian couplings are very difficult to analyse. Hayes and Vigoda, 2003, analysed a coupling for graph colourings with some non-Markovian evolution, but there have been few other attempts.

Other techniques are necessary where coupling seems inapplicable.
Eigenvalue estimates and relatives

From linear algebra, we know that the eventual convergence rate of $\mathcal{M}$ is determined by the gap between the second largest eigenvalue of the matrix $P$, $\lambda_1$, from the largest, 1.

Some techniques attempt to estimate this gap directly. Refinements which better round the initial mixing behaviour are also used.

Various measures have been used: Conductance, Poincaré inequalities, log-Sobolev inequalities, multicommodity flows, local conductance, evolving sets, spectral profile etc.

Unfortunately, these techniques are only applicable to nontrivial problems with the aid of other methods for performing the necessary estimations. The only general techniques for this are canonical paths (and flows) and isoperimetric inequalities.
Eigenvalue estimates and relatives

From linear algebra, we know that the eventual convergence rate of $\mathcal{M}$ is determined by the gap between the second largest eigenvalue of the matrix $P$, $\lambda_1$, from the largest, 1.

Some techniques attempt to estimate this gap directly. Refinements which better round the initial mixing behaviour are also used.

Various measures have been used: Conductance, Poincaré inequalities, log-Sobolev inequalities, multicommodity flows, local conductance, evolving sets, spectral profile etc.

Unfortunately, these techniques are only applicable to nontrivial problems with the aid of other methods for performing the necessary estimations. The only general techniques for this are canonical paths (and flows) and isoperimetric inequalities.
From linear algebra, we know that the eventual convergence rate of $\mathcal{M}$ is determined by the gap between the second largest eigenvalue of the matrix $P$, $\lambda_1$, from the largest, 1.

Some techniques attempt to estimate this gap directly. Refinements which better round the initial mixing behaviour are also used.

Various measures have been used: Conductance, Poincaré inequalities, log-Sobolev inequalities, multicommodity flows, local conductance, evolving sets, spectral profile etc.

Unfortunately, these techniques are only applicable to nontrivial problems with the aid of other methods for performing the necessary estimations. The only general techniques for this are canonical paths (and flows) and isoperimetric inequalities.
Eigenvalue estimates and relatives

From linear algebra, we know that the eventual convergence rate of $\mathcal{M}$ is determined by the gap between the second largest eigenvalue of the matrix $P$, $\lambda_1$, from the largest, 1.

Some techniques attempt to estimate this gap directly. Refinements which better round the initial mixing behaviour are also used.

Various measures have been used: Conductance, Poincaré inequalities, log-Sobolev inequalities, multicommodity flows, local conductance, evolving sets, spectral profile etc.

Unfortunately, these techniques are only applicable to nontrivial problems with the aid of other methods for performing the necessary estimations. The only general techniques for this are canonical paths (and flows) and isoperimetric inequalities.
Conductance

Conductance is one technique for bounding the gap. If the chain $M$ has $\pi$ the uniform distribution, the conductance $\Phi$ is defined to be

$$\Phi = \min_{W \subseteq \Omega, |W| \leq \frac{1}{2} |\Omega|} \frac{\sum_{\omega \in W, \omega' \in \overline{W}} P(\omega, \omega')}{|W|}.$$ 

Good conductance essentially requires that there is no subset of the chain which has a small escape probability.

It may be shown (see, e.g. Sinclair, 1992) that

$$1 - 2\Phi \leq \lambda_1 \leq 1 - \frac{1}{2} \Phi^2.$$ 

The problem is how to estimate $\Phi$ for a large chain.
Conductance

Conductance is one technique for bounding the gap. If the chain $\mathcal{M}$ has $\pi$ the uniform distribution, the conductance $\Phi$ is defined to be

$$
\Phi = \min_{W \subseteq \Omega, |W| \leq \frac{1}{2} |\Omega|} \frac{\sum_{\omega \in W, \omega' \in \overline{W}} P(\omega, \omega')}{|W|}.
$$

Good conductance essentially requires that there is no subset of the chain which has a small escape probability.

It may be shown (see, e.g. Sinclair, 1992) that

$$
1 - 2\Phi \leq \lambda_1 \leq 1 - \frac{1}{2} \Phi^2.
$$

The problem is how to estimate $\Phi$ for a large chain.
Conductance

Conductance is one technique for bounding the gap. If the chain $M$ has $\pi$ the uniform distribution, the conductance $\Phi$ is defined to be

$$\Phi = \min_{W \subseteq \Omega, |W| \leq \frac{1}{2} |\Omega|} \left( \frac{\sum_{\omega \in W, \omega' \in \bar{W}} P(\omega, \omega')}{|W|} \right).$$

Good conductance essentially requires that there is no subset of the chain which has a small escape probability.

It may be shown (see, e.g. Sinclair, 1992) that

$$1 - 2\Phi \leq \lambda_1 \leq 1 - \frac{1}{2} \Phi^2.$$

The problem is how to estimate $\Phi$ for a large chain.
Conductance is one technique for bounding the gap. If the chain $\mathcal{M}$ has $\pi$ the uniform distribution, the conductance $\Phi$ is defined to be

$$
\Phi = \min_{W \subseteq \Omega, |W| \leq \frac{1}{2} |\Omega|} \sum_{\omega \in W, \omega' \in \overline{W}} P(\omega, \omega') \frac{|W|}{|\Omega|}.
$$

Good conductance essentially requires that there is no subset of the chain which has a small escape probability.

It may be shown (see, e.g. Sinclair, 1992) that

$$
1 - 2\Phi \leq \lambda_1 \leq 1 - \frac{1}{2} \Phi^2.
$$

The problem is how to estimate $\Phi$ for a large chain.
Canonical paths

Consider $\mathcal{M}$ as a graph $G = (\Omega, \mathcal{E})$, with $\mathcal{E} = \{(x, y) : P(x, y) > 0\}$, where we suppose all $P(x, y)$ are of size $1/\text{poly}$ in the problem size.

The method routes 1 unit of flow between each pair of states $X, Y$ through $G$ along a specified path. This must be done in such a way that no edge in $\mathcal{E}$ receives more than $C|\Omega|$ flow in total, where $C$ is $\text{poly}$ in the problem size.

Substituting in the expression for $\Phi$ will then give $1/\text{poly}$ conductance, which implies polynomial mixing time.

To prove the property, we show that at any transition from (say) on a path, with the aid of an an additional encoding element $Z$ of $\Omega$ (and a small amount of other information) we can reconstruct $X$ and $Y$. 
Canonical paths

Consider $\mathcal{M}$ as a graph $G = (\Omega, \mathcal{E})$, with $\mathcal{E} = \{(x, y) : P(x, y) > 0\}$, where we suppose all $P(x, y)$ are of size $1/\text{poly}$ in the problem size.

The method routes 1 unit of flow between each pair of states $X, Y$ through $G$ along a specified path. This must be done in such a way that no edge in $\mathcal{E}$ receives more than $C|\Omega|$ flow in total, where $C$ is $\text{poly}$ in the problem size.

Substituting in the expression for $\Phi$ will then give $1/\text{poly}$ conductance, which implies polynomial mixing time.

To prove the property, we show that at any transition from (say) on a path, with the aid of an additional encoding element $Z$ of $\Omega$ (and a small amount of other information) we can reconstruct $X$ and $Y$. 
Canonical paths

Consider \( M \) as a graph \( G = (\Omega, E) \), with \( E = \{(x, y) : P(x, y) > 0\} \), where we suppose all \( P(x, y) \) are of size \( 1/\text{poly} \) in the problem size.

The method routes 1 unit of flow between each pair of states \( X, Y \) through \( G \) along a specified path. This must be done in such a way that no edge in \( E \) receives more than \( C|\Omega| \) flow in total, where \( C \) is \( \text{poly} \) in the problem size.

Substituting in the expression for \( \Phi \) will then give \( 1/\text{poly} \) conductance, which implies polynomial mixing time.

To prove the property, we show that at any transition from (say) on a path, with the aid of an additional encoding element \( Z \) of \( \Omega \) (and a small amount of other information) we can reconstruct \( X \) and \( Y \).
Canonical paths

Consider $\mathcal{M}$ as a graph $G = (\Omega, E)$, with $E = \{(x, y) : P(x, y) > 0\}$, where we suppose all $P(x, y)$ are of size $1/poly$ in the problem size.

The method routes 1 unit of flow between each pair of states $X, Y$ through $G$ along a specified path. This must be done in such a way that no edge in $E$ receives more than $C|\Omega|$ flow in total, where $C$ is $poly$ in the problem size.

Substituting in the expression for $\Phi$ will then give $1/poly$ conductance, which implies polynomial mixing time.

To prove the property, we show that at any transition from (say) on a path, with the aid of an an additional encoding element $Z$ of $\Omega$ (and a small amount of other information) we can reconstruct $X$ and $Y$. 
Example: the hypercube

We will illustrate with a simple example: independent sets in the graph with $n$ vertices and no edges (random walk on the hypercube). We view $\Omega$ as $\{0, 1\}^n$.

The canonical path from $x$ to $y$ will change the coordinates of $x$ to those of $y$ in the order $1, 2, \ldots, n$. At a state $w$ on the path, we are changing position $k$. We see $y$ in positions $1, 2, \ldots, k - 1$, and $x$ in positions $k, k + 1, \ldots, n$.

Clearly, if we take the encoding $z$ to be the vector which is $x$ in positions $1, 2, \ldots, k - 1$, and $y$ in positions $k, k + 1, \ldots, n$, we can reconstruct both $x$ and $y$.

Thus the independent set process will have polynomial mixing time, as we knew from coupling.
Example: the hypercube

We will illustrate with a simple example: independent sets in the graph with $n$ vertices and no edges (random walk on the hypercube). We view $\Omega$ as $\{0, 1\}^n$.

The canonical path from $x$ to $y$ will change the coordinates of $x$ to those of $y$ in the order $1, 2, \ldots, n$. At a state $w$ on the path, we are changing position $k$. We see $y$ in positions $1, 2, \ldots, k - 1$, and $x$ in positions $k, k + 1, \ldots, n$.

Clearly, if we take the encoding $z$ to be the vector which is $x$ in positions $1, 2, \ldots, k - 1$, and $y$ in positions $k, k + 1, \ldots, n$, we can reconstruct both $x$ and $y$.

Thus the independent set process will have polynomial mixing time, as we knew from coupling.
Example: the hypercube

We will illustrate with a simple example: independent sets in the graph with $n$ vertices and no edges (random walk on the hypercube). We view $\Omega$ as $\{0, 1\}^n$.

The canonical path from $x$ to $y$ will change the coordinates of $x$ to those of $y$ in the order $1, 2, \ldots, n$. At a state $w$ on the path, we are changing position $k$. We see $y$ in positions $1, 2, \ldots, k - 1$, and $x$ in positions $k, k + 1, \ldots, n$.

Clearly, if we take the encoding $z$ to be the vector which is $x$ in positions $1, 2, \ldots, k - 1$, and $y$ in positions $k, k + 1, \ldots, n$, we can reconstruct both $x$ and $y$.

Thus the independent set process will have polynomial mixing time, as we knew from coupling.
Example: the hypercube

We will illustrate with a simple example: independent sets in the graph with \( n \) vertices and no edges (random walk on the hypercube). We view \( \Omega \) as \( \{0, 1\}^n \).

The canonical path from \( x \) to \( y \) will change the coordinates of \( x \) to those of \( y \) in the order \( 1, 2, \ldots, n \). At a state \( w \) on the path, we are changing position \( k \). We see \( y \) in positions \( 1, 2, \ldots, k - 1 \), and \( x \) in positions \( k, k + 1, \ldots, n \).

Clearly, if we take the encoding \( z \) to be the vector which is \( x \) in positions \( 1, 2, \ldots, k - 1 \), and \( y \) in positions \( k, k + 1, \ldots, n \), we can reconstruct both \( x \) and \( y \).

Thus the independent set process will have polynomial mixing time, as we knew from coupling.
Zero-one problems

The canonical paths/conductance approach seems to be best suited to sampling from subsets of the unit hypercube. If there are more than $2$ possible values (e.g. $q$-colouring), the reconstruction step seems problematic.

Nevertheless, these methods have led to one of the great successes of MCMC sampling, the estimation of the zero-one permanent. (Jerrum, Sinclair and Vigoda, 2000.)

The other great success, volume of convex bodies, also used conductance, but in a different way, in conjunction with geometric isoperimetry. (Dyer, Frieze and Kannan, 1991)
Zero-one problems

The canonical paths/conductance approach seems to be best suited to sampling from subsets of the unit hypercube. If there are more than $2$ possible values (e.g. $q$-colouring), the reconstruction step seems problematic.

Nevertheless, these methods have led to one of the great successes of MCMC sampling, the estimation of the zero-one permanent. (Jerrum, Sinclair and Vigoda, 2000.)

The other great success, volume of convex bodies, also used conductance, but in a different way, in conjunction with geometric isoperimetry. (Dyer, Frieze and Kannan, 1991)
Zero-one problems

The canonical paths/conductance approach seems to be best suited to sampling from subsets of the unit hypercube. If there are more than 2 possible values (e.g. $q$-colouring), the reconstruction step seems problematic.

Nevertheless, these methods have led to one of the great successes of MCMC sampling, the estimation of the zero-one permanent. (Jerrum, Sinclair and Vigoda, 2000.)

The other great success, volume of convex bodies, also used conductance, but in a different way, in conjunction with geometric isoperimetry. (Dyer, Frieze and Kannan, 1991)
Hardness of approximate counting

Since appropriate counting is no harder than $\text{NP}$, proving a problem hard usually involves showing that, if we can sample, we can solve some known $\text{NP}$-complete problem. This may include optimization approximation problems.

For example, Dyer, Frieze and Jerrum, 1999, showed that, if we can uniformly sample independent sets in graphs with $\Delta \geq 25$, we could approximate the number of satisfied equations in a set of linear equations over $\mathbb{Z}_2$ with 2 variables per equation. This has been shown to be hard by Håstad, 1997.

Dyer, Goldberg, Greenhill and Jerrum, 2003, examined a notion of reducibility amongst approximate counting problems. This suggested that there may be problems, like counting independent sets in bipartite graphs, which are not in $\text{P}$ but are not $\text{NP}$ hard.
Hardness of approximate counting

Since appropriate counting is no harder than $\text{NP}$, proving a problem hard usually involves showing that, if we can sample, we can solve some known $\text{NP}$-complete problem. This may include optimization approximation problems.

For example, Dyer, Frieze and Jerrum, 1999, showed that, if we can uniformly sample independent sets in graphs with $\Delta \geq 25$, we could approximate the number of satisfied equations in a set of linear equations over $\mathbb{Z}_2$ with 2 variables per equation. This has been shown to be hard by Håstad, 1997.

Dyer, Goldberg, Greenhill and Jerrum, 2003, examined a notion of reducibility amongst approximate counting problems. This suggested that there may be problems, like counting independent sets in bipartite graphs, which are not in $\text{P}$ but are not $\text{NP}$ hard.
Hardness of approximate counting

Since appropriate counting is no harder than $\text{NP}$, proving a problem hard usually involves showing that, if we can sample, we can solve some known $\text{NP}$-complete problem. This may include optimization approximation problems.

For example, Dyer, Frieze and Jerrum, 1999, showed that, if we can uniformly sample independent sets in graphs with $\Delta \geq 25$, we could approximate the number of satisfied equations in a set of linear equations over $\mathbb{Z}_2$ with 2 variables per equation. This has been shown to be hard by Håstad, 1997.

Dyer, Goldberg, Greenhill and Jerrum, 2003, examined a notion of reducibility amongst approximate counting problems. This suggested that there may be problems, like counting independent sets in bipartite graphs, which are not in $\text{P}$ but are not $\text{NP}$ hard.