Approximate Counting, Almost Uniform Generation and Random Walks

Michal Parnas
Department of Computer Science
Hebrew University
Jerusalem, Israel

August, 1989

A thesis submitted in fulfillment
of the requirements for the degree of
Master of Science

supervised by
Prof. Avi Wigderson

Contents

1 Introduction

Counting is the process of finding the number of elements of a given set, for example counting the number of students in a given class, or finding the number of people that voted Likud in the last elections. The obvious way to solve problems of this type, is by exhaustive search of the set of interest. In the first example this may take only a few seconds, but in the second it may take a couple of days (depending on the computational power). One may try to develop more efficient counting procedures which depend on the special structure of the set of interest. In the first example, one can count the number of chairs in a row, and multiply it by the number of rows. It is possible that the answer will not be exact, because some of the seats may be unoccupied. In the election example, one may sample from the set of all votes, get an estimation of the percentage of Likud voters, and multiply that percentage by the number of all voters (assuming that number is known). The answer will only be an estimate. The quality of the estimate depends naturally on the number of samples made, and on the distribution of the samples over the total population.

In 1979, Valiant [V] defined the class $\#P$, of counting problems which are believed to be difficult, i.e. are not known to be solvable in polynomial time. For problems which belong to this class it is natural to seek approximate solutions, in order to reduce the running time of the algorithm. One type of existing approximate counting algorithms are the Monte-Carlo algorithms. They are randomized algorithms which estimate the size of a set with some performance guarantee, i.e. the answer deviates by no more then a specified factor from the exact solution. The running time of the algorithm depends on this factor, and is usually faster than any known deterministic algorithm for the problem. We will review the Monte-Carlo technique in section 2, and bring some examples of counting problems solved using this technique in the past few years.

The example of counting electoral votes, raises the idea that there may be a connection between counting the number of elements of a given set and sampling from it. Jerrum, Valiant and Vazirani [JVV], observed this connection, and defined the class *Uniform Generation*. The algorithms in this class generate (sample) uniformly at random the elements of a given set. [JVV] proved that this class is reducible to counting. That is, given a deterministic (approximate) counting algorithm for a given set, it is possible to build a uniform generator for the elements of that set. It was also shown that the classes of *randomized approximate counting* algorithms and *almost uniform generation* algorithms are equivalent, for problems which are *self reducible*. This results are described in sections 3 and 4.

Given the above results it was understood, that it may be easier to construct almost uniform generators and use them as building blocks in randomized approximate counting algorithms. Broder [B], was the first to use Markov chains as almost uniform generators. The main idea is to define a Markov chain on the set, whose size we want to estimate, then to simulate the chain until it is close to its stationary distribution, and output the state we have reached as the output of an almost uniform generator. Broder defined an ergodic Markov chain on the set of all perfect and almost perfect matchings in a dense bipartite graph. He showed that if it converges fast enough to its stationary distribution then it can be used as an almost uniform generator for perfect matchings. Jerrum and Sinclair [JS], were the first to prove that the chain Broder defined converges fast (see section 8). The Markov chain method is described in section 5.

The two major problems which arise when using Markov chains as almost uniform generators are:

- What is the rate of convergence of the chain?
- What is the stationary distribution of the chain?

Polynomial rate of convergence is necessary so that the method will also be practical. In section 6, we will review two methods to determine the rate of convergence of a given Markov chain: Coupling by Aldous [A], and Conductance by Jerrum and Sinclair [SJ].

The second question has no easy solution. For reversible Markov chains it will be shown that it is possible to compute the stationary distribution of each state, as the simulation of the chain proceeds (see section 5). In the general case we have to try and bound the stationary distribution. If we can bound the ratio of the stationary distribution of two neighbor states, then we can derive a global bound on the ratio of the stationary distribution of any two states, as a function of the diameter of the *underlying graph* of the chain (see sections 5 and 9.2).

We will illustrate the above concepts by studying the random walk on the n-dimensional cube over Z_k. Of course the number of vertices of the cube is known exactly, and therefore there is no need to estimate it, but the rate of convergence of this walk is of independent interest for itself. We will show that the walk converges rapidly to a uniform stationary distribution, using the Coupling technique (see section 7). As far as we know, no other proof for the rapid convergence of this walk was known before.

The next problem which shall be studied, is counting the number of independent sets in a graph (section 9). This problem was shown to be $\#P$-complete [PB], and therefore is not likely to have an exact solution, but an approximate one may very well be possible. Recently, Dagum showed that it is possible to estimate the number of independent sets of a claw free graph in polynomial time ([D]). He used technics similar to those used by Jerrum and Sinclair ([SJ],[JS]), to bound the conductance of the *exchange graph*. The problem of estimating the number of independent sets of a general graph, (not necessarily claw free) in polynomial time, remains open. We will try to use the coupling method, in order to bound the rate of convergence of more direct walks on the set of independent sets of a given graph. For the simple case of a graph with degree ≤ 2, a Markov Chain with uniform stationary distribution, will be defined on the set of independent sets of the graph, and will be shown to converge rapidly. For the general case, a similar Markov Chain will be introduced, which again converges fast, but to an unknown stationary distribution. Proof that the stationary distribution is bounded by a polynomial for a certain class of graphs, will immediately yield a polynomial randomized approximate counting algorithm for independent sets of graphs from that class.

For a graph which is a tree, a deterministic algorithm for counting both the number of independent sets and maximum independent sets of the graph will be presented, which runs in polynomial time (section 10).

Finally we will examine the problem of finding the number of partitions of a given integer n. An efficient deterministic algorithm for this problem is not known. We will try to construct an approximate one, using the Markov chain method. We will define two Markov chains - one on the set of partitions of n into k parts, and the second on the set of all partitions of n. Both chains will be shown to be reversible and rapidly converging. Their stationary distribution will be computed exactly. However since it is not bounded polynomialy in n, it can not be used to estimate the number of partitions of n in polynomial time. As a possible solution, a different Markov chain will be defined on the set of partitions of n into k parts. It has a stationary distribution which is bounded by a polynomial in n and k, however its rate of convergence is unknown so far (section 11).

2 Monte - Carlo Algorithms for Counting Problems

Enumeration problems can be described as computing the size of a set of elements S. We shall say $|S|$ is difficult to compute, if any known deterministic algorithm which finds $|S|$, runs in time proportional to $|S|$. For small sets S this may not be too bad, but for exponential size sets the computation may take quite a long time. In this section we will describe a class of algorithms which estimate the size of a set S, such that the running time of the algorithm is faster then any known deterministic algorithm which computes $|S|$, and depends on the accuracy of the estimation.

We first need to define when an estimate is "accurate enough". The following definition is due to Karp and Luby [KL]:

Definition 2.1 A random variable Y is called an (ϵ, δ)-*approximation* to Q if

$$Pr((1+\epsilon)^{-1}Q \leq Y \leq (1+\epsilon)Q) \geq 1-\delta$$

Monte - Carlo algorithms are a randomized computational method for the estimation of some quantity Q. In general, a Monte-Carlo algorithm to estimate Q consists of N independent trials of the same experiment, such that if X_i is the outcome of the i'th trial then $E(X_i) = Q$. The output of the algorithm is $Y = \frac{X_1+...+X_N}{N}$. It is clear that $E(Y) = Q$, and the question is, what is the number of experiments N, which guaranty that Y is a good estimate of Q.

Lemma 2.1 *Let $X_1, ..., X_N$ be N independent random variables with expectation $\mu = E(X_i)$, such that $0 \leq X_i \leq z$, and let $Y = \frac{X_1+...+X_N}{N}$. Then Y is an $(\epsilon-\delta)$- approximation to μ if $N \geq \frac{z}{\mu\epsilon^2}\log\frac{2}{\delta}$ for $\epsilon \ll \mu$.*

Proof: By the Hoeffding inequality (see [Ho]),

$$Pr(Y \geq \mu+t) \leq [(\frac{\mu}{\mu+t})^{\frac{\mu+t}{z}}(\frac{z-\mu}{z-\mu-t})^{1-\frac{\mu+t}{z}}]^N$$

for all $0 \leq t \leq z-\mu$.
Take $t = \epsilon\mu$, then

$$Pr(Y \geq (1+\epsilon)\mu) \leq [(\frac{1}{1+\epsilon})^{(1+\epsilon)\mu}(1+\frac{t}{z-\mu-t})^{z-\mu-t}]^{\frac{N}{z}} \leq e^{\frac{-\epsilon^2 N\mu}{z}}$$

for $\epsilon \ll \mu$. In a similar way it is possible to prove that

$$Pr(Y \leq (1+\epsilon)^{-1}\mu) \leq e^{\frac{-\epsilon^2 N\mu}{z}}$$

Take $N \geq \frac{z}{\mu\epsilon^2}\log\frac{2}{\delta}$. Then

$$Pr((1+\epsilon)^{-1}\mu \leq Y \leq (1+\epsilon)\mu) \geq (1-\frac{\delta}{2})(1-\frac{\delta}{2}) \geq 1-\delta$$

Hence Y is an $(\epsilon - \delta)$-approximation to μ. $\square$

Notice that the restriction $\epsilon \ll \mu$, is not a real problem, since we are interested in estimating the size of large sets, hence $\mu = Q$ is large and of course $\epsilon < 1$, therefore the requirement $\epsilon \ll \mu$ is satisfied.

A slightly different bound on the number of trials N, needed to guaranty a good estimate is proved in the following lemma.

Lemma 2.2 *Let $X_1, ..., X_N$ be N independent random variables, with expectation $E(X_i) = \mu$. Then $N \geq \frac{E(X_i^2)}{E^2(X_i)} \frac{36}{\epsilon^2} \log \frac{1}{\delta}$ is sufficient to guaranty an (ϵ, δ)-approximation to μ.*

Proof: Let $Y = \frac{X_1 + ... + X_n}{n}$. Using the Chebychev Inequality, it is easy to prove that Y is an $(\epsilon, \frac{1}{4})$-approximation to μ, for $n \geq \frac{E(X_i^2)}{E^2(X_i)} \frac{4}{\epsilon^2}$. Jerrum, Valiant and Vazirani show in [JVV], how to get an (ϵ, δ)- approximation of μ for any δ. They suggest to make $t = 12 \log \frac{1}{\delta}$ estimates Y, each an $(\epsilon - \frac{1}{4})$-approximation to μ, and to take their median as a final estimate to μ. It is possible to show, using the Chernoff inequality, that this estimate is an $(\epsilon - \delta)$-approximation to μ. The complexity of this algorithm is $N = tn \geq \frac{E(X_i^2)}{E^2(X_i)} \frac{36}{\epsilon^2} \log \frac{1}{\delta}$. $\square$

Let us now review some counting problems, which were solved using the Monte-Carlo method. We will use either one of the above two lemmas, when trying to bound the number of trials needed in a Monte-Carlo algorithm.

2.1 Estimating the Weight of a Set

Let R be a set, and assume there is a weight function $w : R \to \Re$, which assigns a weight $w(x)$ to each $x \in R$. We would like to estimate the weight of R, that is $w(R) = \sum_{x \in R} w(x)$. The following Monte-Carlo algorithm which estimates $w(R)$, is due to [KLM]. The algorithms described in the next subsections, for more specific counting problems will be based on this algorithm.

Suppose we can find a set S, such that:

1. $R \subseteq S$.
2. $w(S)$ is known, or can be computed in polynomial time.
3. There exists a polynomial sampling algorithm which draws an element $x \in S$ with probability $w(x)/w(S)$.
4. There exists a polynomial algorithm which answers for any x sampled, whether $x \in R$.
5. The ratio $w(S)/w(R)$ is small (polynomial in $\log |R|$).

Then we can design a Monte-Carlo algorithm which estimates $|R|$ in polynomial time in $\log|R|$ (or in $|x|$):

Algorithm:

1. Draw N elements x from S.
2. For each x, set $X_i = w(S)$ if $x \in R$, and $X_i = 0$ otherwise.
3. Estimate $w(R)$ by $Y = \frac{X_1+...+X_N}{N}$.

The triple (S, R, w) is called a *Monte - Carlo structure* with universe S, target R and weight function w.

From lemma 2.1, we get a bound on the number of trials needed in order to guaranty that Y is a good estimate to $w(R)$:

Theorem 2.1 *If $N \geq \frac{w(S)}{w(R)} \frac{1}{\epsilon^2} \log \frac{2}{\delta}$ then Y is an (ϵ, δ)-approximation to $w(R)$.*

Proof: Simple calculations show that $E(Y) = E(X_i) = w(R)$, and $0 \leq X_i \leq w(S)$. □

As a result we can see that the efficiency of the method is determined by the ratio $\frac{w(S)}{w(R)}$, and therefore finding a smaller set S will result in a more efficient algorithm.

2.2 Estimating the Relative Size of a Subset

Assume we are given a subset $R \subseteq S$, and would like to estimate the relative size of R compared to S, that is the ratio $\frac{|R|}{|S|}$. Jerrum, Valiant and Vazirani in [JVV], and Broder in [B], showed how to estimate $\frac{|R|}{|S|}$, when we are given a uniform sampling algorithm for elements of S. In this case the problem is only a private case of the problem described in section 2.1. We will expand their idea for a general sampling algorithm, which draws an element $x \in S$ with probability p_x (not necessarily uniform), provided that the sampling probabilities are known in advance.

Algorithm:

1. Draw N elements $x_1, ..., x_N$ from S.
2. Set $X_i = \frac{1}{p_{x_i}}$ if $x_i \in R$, $X_i = 0$ else, and define $X = \frac{X_1 + ... + X_N}{N}$.
3. Set $Y_i = \frac{1}{p_{x_i}}$, and $Y = \frac{Y_1 + ... + Y_N}{N}$.
4. Estimate $\frac{|R|}{|S|}$ by $\frac{X}{Y}$.

We shall need the following two technical lemmas to show that $\frac{X}{Y}$ is a good estimate to $\frac{|R|}{|S|}$:

Lemma 2.3 *1. $E(X) = E(X_i) = |R|$.*

2. $E(Y) = E(Y_i) = |S|$.

Lemma 2.4 *If x and y are an (ϵ', δ')-approximation to x' and y' respectively, then $\frac{x}{y}$ is an (ϵ, δ)-approximation to $\frac{x'}{y'}$, where $\epsilon = 2\epsilon' + \epsilon'^2$ and $\delta = 2\delta' - \delta'^2$.*

Proof: It is easy to verify that if y is an (ϵ', δ')-approximation to y', then y' is an (ϵ', δ')-approximation to y. Hence:

$$Pr((1+\epsilon)^{-1}\frac{x'}{y'} \leq \frac{x}{y} \leq \frac{x'}{y'}(1+\epsilon)) =$$
$$Pr((1+\epsilon')^{-2}\frac{x'}{y'} \leq \frac{x}{y} \leq \frac{x'}{y'}(1+\epsilon')^2) =$$
$$Pr((1+\epsilon')^{-2}x'y \leq xy' \leq x'y(1+\epsilon')^2) \geq$$
$$Pr((1+\epsilon')^{-1}x' \leq x \leq x'(1+\epsilon')) \cdot Pr((1+\epsilon')^{-1}y \leq y' \leq y(1+\epsilon')) \geq$$
$$(1-\delta')(1-\delta') = 1-\delta$$

$\square$

Define

$$P_R = \max_{x \in R} \frac{1}{p_x}$$

and

$$P_S = \max_{x \in S} \frac{1}{p_x}$$

Then:

Theorem 2.2 *If $N \geq \max\{\frac{P_R}{|R|}\frac{1}{\epsilon'^2}\log\frac{2}{\delta'}, \frac{P_S}{|S|}\frac{1}{\epsilon'^2}\log\frac{2}{\delta'}\}$ then $\frac{X}{Y}$ is an (ϵ, δ)- approximation to $\frac{|R|}{|S|}$, for $\epsilon = 2\epsilon' + \epsilon'^2$ and $\delta = 2\delta' - \delta'^2$.*

Proof: From lemma 2.1 and lemma 2.3:

- X is an (ϵ', δ')-approximation to $|R|$, for $N \geq \frac{P_R}{|R|}\frac{1}{\epsilon'^2}\log\frac{2}{\delta'}$.
- Y is an (ϵ', δ')-approximation to $|S|$, for $N \geq \frac{P_S}{|S|}\frac{1}{\epsilon'^2}\log\frac{2}{\delta'}$.

Take $N = \max\{\frac{P_R}{|R|}\frac{1}{\epsilon'^2}\log\frac{2}{\delta'}, \frac{P_S}{|S|}\frac{1}{\epsilon'^2}\log\frac{2}{\delta'}\}$. Then by lemma 2.4, $\frac{X}{Y}$ is an (ϵ, δ)- approximation to $\frac{|R|}{|S|}$. □

For the simple case, of a uniform sampling algorithm we get:

Corollary 2.1 *If $p_x = \frac{1}{|S|}$ for every $x \in S$, then for $N \geq \frac{|S|}{|R|}\frac{1}{\epsilon'^2}\log\frac{2}{\delta'}$, $\frac{X}{Y}$ is an (ϵ, δ) approximation to $\frac{|R|}{|S|}$, where $\epsilon = 2\epsilon' + \epsilon'^2$ and $\delta = 2\delta' - \delta'^2$.*

We can see, that if we are given a uniform sampling algorithm, then the number of trials needed to guaranty a good approximation, depends on the ratio $\frac{|S|}{|R|}$, and thus we need at least a crude estimate of this ratio in order to get a good estimate of its inverse. For the general case, the number of trials depends on the deviation of the given sampling algorithm from a uniform sampling algorithm. The number of trials will be polynomial if the ratios $\frac{P_R}{|R|}$ and $\frac{P_S}{|S|}$ are polynomial.

If we know what $|S|$ is, then we can estimate $|R|$, by multiplying the ratio $\frac{X}{Y}$ by $|S|$. Thus we get:

Corollary 2.2 *If $N \geq \max\{\frac{P_R}{|R|}\frac{1}{\epsilon'^2}\log\frac{2}{\delta'}, \frac{P_S}{|S|}\frac{1}{\epsilon'^2}\log\frac{2}{\delta'}\}$ then $|S|\frac{X}{Y}$ is an (ϵ, δ)- approximation to $|R|$, for $\epsilon = 2\epsilon' + \epsilon'^2$ and $\delta = 2\delta' - \delta'^2$.*

2.3 DNF Formulas

We would like to estimate the number, $\#F$, of assignments that satisfy a DNF formula F. This problem was shown to be $\#P$ ([V]), hence probably does not have a fast deterministic algorithm. Karp and Luby in [KL], found a polynomial Monte-Carlo algorithm for the problem, which we will describe now.

Assume the number of literals in F is n, and the number of clauses is m.

Following the algorithm for estimating the weight of a set, let -

$$S = \{(j, x) | x \textit{ is an assignment satisfying the } j'th \textit{ clause of } F\}$$

$$R = \{(j, x) | \textit{ the } j'th \textit{ clause is the lowest numbered clause satisfied by } x\}$$

$$E_j = \{x | x \text{ } is \text{ } an \text{ } assignment \text{ } satisfying \text{ } the \text{ } j'th \text{ } clause \text{ } of \text{ } F\}$$

Then $|S| = \sum_{j=1}^{m} |E_j|$ and $|R| = |\bigcup_{j=1}^{m} E_j| = \#F$.

Algorithm:

1. Compute $|S|$ - If clause j contains no pair of literals x_l and $\neg x_l$, then $|E_j| = 2^{n-k}$, where k is the number of literals in clause j. Else $|E_j| = 0$. Thus $|S|$ is easily determined in time $O(mn)$.

2. Make N samples from S, each one as follows- Select clause j with probability $|E_j|/|S|$. Given j, assign to each x_k and $\neg x_k$ in clause j the values 'true' and 'false' respectively. If neither x_k or $\neg x_k$ occur in clause i then assign a random independent assignment to x_k. Each of the elements of S is equally likely to be selected by this process, in time $O(n)$.

3. Check whether assignment x selected above to satisfy clause j, satisfies any clause j', $j < j'$. If not then $(x, j) \in R$, and therefore set $X_i = |S|$; else $(x, j) \notin R$, hence set $X_i = 0$. The number of steps required is $O(mn)$.

4. Estimate $|R| = \#F$ by $Y = \frac{X_1 + \ldots + X_N}{N}$.

Again we get the following bound on the number of trials needed to guaranty a good estimate:

Theorem 2.3 *If $N \geq m\frac{1}{\epsilon^2}\log\frac{2}{\delta}$, then Y is an (ϵ, δ)-approximation to $\#F$.*

Proof: By lemma 2.1, since $E(Y) = E(X_i) = |R|$, $0 \leq X_i \leq |S|$, and $\frac{|S|}{|R|} \leq m$. □

The whole algorithm can be executed in $O(m^2 n \frac{1}{\epsilon^2}\log\frac{1}{\delta})$, since each trial takes $O(mn)$. [KLM] show a faster way to perform step 3. above, thus reducing the total time of the algorithm to $O(mn\frac{1}{\epsilon^2}\log\frac{1}{\delta})$.

2.4 The Multiterminal Network Reliability - MNR - Problem

Let us review another important example, of determining the failure probability of a network, whose edges may fail at random. The general problem was shown to be $\#P$-complete ([PB]). Karp and Luby devised a Monte-Carlo algorithm for this problem. We will bring the main aspects of this problem here. For a full description the reader is referred to [KL].

A Multiterminal Network (G, K) is :

1. a connected undirected graph $G = (V, E)$.

2. a set $K \subseteq V$ of terminals.

3. a failure probability $0 < P_e < 1$ assigned to each $e \in E$.

An edge $e \in E$ is *bad* with probability P_e and *good* with probability $1 - P_e$. A *state* of the network is a function $s : E \to \{good, bad\}$. The probability of state s is $b(s) = \prod_{\{e|s(e)=bad\}} P_e \prod_{\{e|s(e)=good\}}(1 - P_e)$. A state s is a *failure state* if for some pair $x, y \in K$ every path between x and y contains an edge z such that $s(z) = bad$. The *failure probability of the network* is $b(F) = \sum_{s \in F} b(s)$, where F is the set of failure states.

Considering the above, the *MNR Problem* is to estimate the failure probability of the network.

A *cut* is a minimal set of edges, which when removed from the connected graph G, results in a disconnected graph $G' = G_1 \dot\cup G_2$. A cut c separates x and y if $x \in G_1$ and $y \in G_2$. A cut is K - *separating* if it separates two vertices in K.

Lemma 2.5 *State s of (G, K) is a failure state iff there is a K - separating cut c, such that for each $z \in c, s(z) = bad$. The cut c is called a failing cut.*

We want to define a specific cut $c(s)$, for each failure state s, so that we can identify s by its cut $c(s)$:

Let C_i be the vertices of the connected component of the graph induced by G by including only the good edges, such that the i'th terminal is included. Let i be the least index for which the subgraph of G induced by all vertices not in C_i is connected. Choose $c(s)$ as the K-separating cut consisting of those edges having exactly one of their two endpoints in C_i.

2.4.1 A Monte - Carlo algorithm based on cuts for estimating $b(F)$

The Monte-Carlo algorithm Karp and Luby describe, which estimates $b(F)$, is based on the algorithm sketched in section 2.1, and proceeds as follows:
Let

$$S = \{(s, c) | s \text{ is a failure state}, \ c \text{ is a failing cut}\}$$

$$R = \{(s, c(s)) | s \text{ is failure state}\}$$

and

$$w((s, c)) = b(s)$$

(S, R, w) is a Monte Carlo structure based on cuts for the network (G, K), where $w(R) = b(F)$.

Algorithm:

1. $w(S) = \sum_c \prod_{e \in c} P_e$ where c is a K - separating cut.

2. Draw N states $s \in S$, each one as follows - choose a K - separating cut c, with probability $\prod_{e \in c} \frac{P_e}{w(S)}$. Choose a state s as follows -

   - If $e \in c$ then $s(e) = bad$.
   - If $e \notin c$ then $s(e) = bad$ with probability P_e.

3. Test whether (s, c) is in R - compute $c(s)$ and test whether $c = c(s)$. If so set $X_i = w(S)$, else $X_i = 0$.

4. Estimate $b(F)$ by $Y = \frac{X_1 + ... + X_N}{N}$.

Theorem 2.4 *If $N \geq \prod_{e \in E}(1 + P_e)\frac{1}{\epsilon^2}\log\frac{2}{\delta}$, then Y is an (ϵ, δ)-approximation to $b(F)$.*

Proof: Follows from lemma 2.1, since $w(S)/w(R) \leq \prod_{e \in E}(1 + P_e)$ (see [KL]). □

Thus, if the failure probability of the edges is small enough, then a small N will guarantee a good approximation. Notice, that this algorithm will be efficient only if a list of all K-separating cuts is available. Without such a list it would be difficult to compute $w(S)$, or to sample from S. Karp and Luby overcome these difficulties by constructing efficient Monte-Carlo algorithms for estimating $b(F)$, which do not require a list of the cuts, but assuming the graph G is planar (see [KL]).

2.5 Translates

Let us consider another simple example. Here a fast deterministic algorithm is known, but the Monte-Carlo algorithm we shall describe shortly is even faster, and thus worthwhile examining. The algorithm can be found in [KLM].

Let $a = \{a_1, ..., a_m\}$ be a set of m distinct points in the plane with integer coordinates $(a_k(x), a_k(y))$ for $k = 1, ..., m$. A *translate* of a is a pair of integers $(\Delta x, \Delta y)$, which maps a to a set of points a', where $a'_k(x) = a_k(x) + \Delta x$ and $a'_k(y) = a_k(y) + \Delta y$ for $k = 1, ..., m$.

Let $b = \{b_1, ..., b_m\}$ be another set of distinct points in the plane, and let

$$T = \{(\Delta x, \Delta y) | \exists k, j \ a_k(x) + \Delta x = b_j(x) \ and \ a_k(y) + \Delta y = b_j(y)\}$$

We want to find $|T|$. A deterministic algorithm can find $|T|$ in $O(m^2 logm)$, by first computing all pairs $(b_j(x) - a_k(x), b_j(y) - a_k(y))$ for $k, j = 1, ..., m$, and then sorting these values to eliminate duplicates.

2.5.1 A Monte - Carlo algorithm for estimating $|T|$

Let

$$S = \{(k,j)|k,j = 1,...,m\}, \quad |S| = m^2$$

and let

$$T_k = \{(b_j(x) - a_k(x), b_j(y) - a_k(y))|j = 1,...,m\}$$

Then $T = \bigcup_{k=1}^{m} T_k$, $m \le |T| \le m^2$ and $|S|/|T| \le m$.

Algorithm:

1. Sort the set b lexicographicly, in time $O(m \log m)$ (any sorting algorithm with a running time of $O(m \log m)$ will do).
2. Perform N trials, where each trial will consist of choosing randomly three indices h, j, k and checking whether $(b_j(x) - a_h(x), b_j(y) - a_h(y)) \in T_k$. Since the set b is sorted, the above check can be carried out using binary search, to find if $\exists l$ such that $(a_k(x) + b_j(x) - a_h(x), a_k(y) + b_j(y) - a_h(y)) = (b_l(x), b_l(y))$. If so then set $X_i = |S|$, else $X_i = 0$.
3. Estimate $|T|$ by $Y = \frac{X_1 + ... + X_N}{N}$.

Theorem 2.5 *If $N \ge m\frac{1}{\epsilon^2}\log\frac{2}{\delta}$, then Y is an (ϵ,δ)-approximation to $|T|$.*

Proof: Follows directly from lemma 2.1, since $\frac{|S|}{|T|} \le m$. □

Thus, the whole algorithm can be executed in $O(m \log m \frac{1}{\epsilon^2}\log\frac{1}{\delta})$, because each trial takes $O(logm)$ steps.

2.6 The Permanent

We will conclude this section with a final important example, which we will incounter again in section 8.

Let A be an $n * n$ matrix over $\Sigma = \{0,1\}$. Then $Per(A) = \sum_{\sigma \in S_n} a_{1\sigma(1)}.....a_{n\sigma(n)}$ is the permanent of A. It is easily verified that $Per(A)$ is the number of perfect matchings in the bipartite graph $G = (V_1, V_2, E)$, where $|V_1| = |V_2| = n$ and $e = (i,j) \in E$ iff $a_{ij} = 1$.

Many efforts have been made to find an efficient algorithm for this problem, but so far the fastest known deterministic algorithm runs in time $O(n2^n)$, and is due to Ryser ([Ry]). Furthermore, it has been shown that computing the permanent exactly is $\#P$-complete ([V]). We will now bring a Monte-Carlo algorithm developed by [KKLLL], for this problem. Its running time is still exponential, but it is faster then the running time of Ryser's algorithm.

2.6.1 A simple Monte-Carlo algorithm for Per(A)

Algorithm:

1. Define N random matrices $B_i = (b_{jk})$, where $b_{jk} = \pm a_{jk}$ with probability $1/2$.
2. Set $X_i = det(B_i)^2$.
3. Estimate $Per(A)$ by $Y = \frac{X_1+...+X_N}{N}$.

As we saw in the previous examples, we have to show now that $E(Y) = E(X_i) = Per(A)$, and to find the number of trials N, needed to guaranty a good estimate. We will start with the first:
Let

$$P = \{\sigma | a_{j\sigma(j)} = 1 \quad for \ j = 1, ..., n\}$$

the set of permutations which contribute one to the permanent.

Theorem 2.6 *$E(X_i) = Per(A)$.*

Proof:

$$X_i = det(B_i)^2 = (\sum_{\sigma \in S_n} sgn(\sigma) \prod_{j=1}^{n} b_{j\sigma(j)})^2 =$$

$$\sum_{\sigma \in P} (\prod_{j=1}^{n} b_{j\sigma(j)})^2 + \sum_{\sigma_k, \sigma_l \in P, \ k \neq l} (sgn(\sigma_k \sigma_l) \prod_{j=1}^{n} b_{j\sigma_k(j)} b_{j\sigma_k(l)})$$

- The first term in the sum equals $Per(A)$.
- For each two different permutations σ_k and σ_l, there exist at least two indices s, t, for which $\sigma_k(s) = b_{st} \neq \sigma_l(s)$. Hence we can express each summand in the second term of the sum, as a product of b_{st} and of a function $H(\sigma_k, \sigma_l)$ which does not contain b_{st}. But $E(b_{st}) = 0$, and therefore the expected value of the second term of the sum is 0.

$\square$

We now have to find for what N, the estimate Y, ia an (ϵ, δ)- approximation to the permanent. From lemma 2.2 we see that we have to bound $\frac{E(X_i^2)}{E(X_i)^2}$, and this will be achieved through the following two lemmas:

Label each pair $< j, \sigma(j) >$ with σ, and for each r-tuple $< \sigma_1, ..., \sigma_r >$ define a graph $G(\sigma_1, ..., \sigma_r)$ where -

- There is a node for each $< j, k >$ labelled with one or more of $\sigma_1, ..., \sigma_r$.
- There is an edge between $< j, k >$ and $< j', k' >$ iff $j = j'$ or $k = k'$.

Let $c(G)$ be the number of cycles of length ≥ 2 in graph G.
Let

$$D_2 = \{G(\sigma_1, \sigma_2) | < \sigma_1, \sigma_2 > \in P^2\}$$

the set of graphs which are defined by two permutations in P. For each $G \in D_2$ let

$$[G]_2 = \{< \sigma_1, \sigma_2 > \in P^2 | G(\sigma_1, \sigma_2) = G\}$$

the set of pairs of permutations in P, which define the same graph G.

Lemma 2.6 $[G]_2 = 2^{c(G)}$.

Proof: Let $< \sigma_1, \sigma_2 > \in [G]_2$. Then each cycle of length one in G is labelled with both σ_1 and σ_2. Let c be a cycle of length ≥ 2 in G. Every node at an even distance from a fixed node in c, should be labelled with the same label, and every other node with the other label. Thus there are two possible labellings for each cycle of length ≥ 2, and therefore a total of $2^{c(G)}$ labellings. □

Define

$ODD = \{< \sigma_1, ..., \sigma_4 > | \exists < s, t > \ \textit{labelled with an odd number of labels from } \sigma_1, ..., \sigma_4\}$

$EVEN = P^4 - ODD$
and let

$$D_4 = \{G(\sigma_1, ..., \sigma_4) | < \sigma_1, ..., \sigma_4 > \in EVEN\}$$

the set of graphs which are defined by a set of permutations from $EVEN$. Then it is easy to check that $D_4 = D_2$. For each $G \in D_4$ let

$$[G]_4 = \{< \sigma_1, ..., \sigma_4 > \in EVEN | G(\sigma_1, ..., \sigma_4) = G\}$$

the set of all permutations from $EVEN$ which define the same graph G.

Lemma 2.7 $[G]_4 = 6^{c(G)}$.

Proof: Let $<\sigma_1, ..., \sigma_4> \in [G]_4$. Each cycle of length one in G is labelled with all four of $\sigma_1, ..., \sigma_4$. Let c be a cycle of length ≥ 2 in G. Every node at an even distance from c should be labelled with the same two permutations out of $\sigma_1, ..., \sigma_4$, and every node at an odd distance with the remaining two. Thus there are $\binom{4}{2} = 6$ possible labellings for each cycle of length ≥ 2, and therefore a total of $6^{c(G)}$ labellings. □

And finally,

Lemma 2.8 $\frac{E(X_i^2)}{E(X_i)^2} \leq 3^{n/2}$.

Proof:

1. $E(X_i)^2 = Per(A)^2 = \sum_{P^2} 1 = \sum_{G \in D_2} \sum_{[G]_2} 1 = \sum_{G \in D_2} 2^{c(G)}$ (by lemma 2.6).
2. Let $X(\sigma_1, ..., \sigma_4) = \prod_{k=1}^{4} (sgn(\sigma) \prod_{j=1}^{n} b_{j\sigma_k(j)})$. Then:

$$X_i^2 = det(B_i)^4 = \sum_{P^4} X(\sigma_1, ..., \sigma_4) = \sum_{ODD} X(\sigma_1, ..., \sigma_4) + \sum_{EVEN} X(\sigma_1, ..., \sigma_4)$$

   - For each four permutations $\sigma_1, \sigma_2, \sigma_3, \sigma_4 \in ODD$, there exist two indices s, t which are labelled with an odd number of labels from these four permutations. Thus either it is labelled by only one, (say σ_1) or it is labelled by three and not labelled by one (say again σ_1). Hence $\sigma_1(s) = t \neq \sigma_2, \sigma_3, \sigma_4$. Therefore we can express each summand in the sum over ODD, as $H(\sigma_1, ..., \sigma_4)b_{st}$, where H is a function which does not contain b_{st}. But $E(b_{st}) = 0$, and therefore the expected value of the first term is 0.
   - For each summand in the sum over $EVEN$, we have that $X(\sigma_1, ..., \sigma_4) = 1$, independent of B_i, and thus $E(X(\sigma_1, ..., \sigma_4)) = 1$.

   Thus $E(X_i^2) = \sum_{EVEN} 1 = \sum_{G \in D_4} \sum_{[G]_4} 1 = \sum_{G \in D_4} 6^{c(G)}$ (by lemma 2.7).

Therefore $\frac{E(X_i^2)}{E(X_i)^2} = \frac{\sum_{D_4} 6^{c(G)}}{\sum_{D_2} 2^{c(G)}} \leq max_G 3^{c(G)} \leq 3^{n/2}$, because there are at most $n/2$ cycles of length ≥ 2 in any graph G. □

As a final result we get:

Theorem 2.7 *It is possible to get an $(\epsilon - \delta)$-approximation to $Per(A)$, if $N = 3^{n/2} \frac{36}{\epsilon^2} \log \frac{1}{\delta}$.*

Proof: Follows from lemma 2.2 and lemma 2.8. $\square$

Therefore the total running time of the algorithm is $3^{n/2}\frac{36}{\epsilon^2}\log\frac{1}{\delta}poly(n)$, where $poly(n)$ is the time each trial takes.

2.6.2 A modified Monte-Carlo algorithm for Per(A)

[KKLLL] improve the bound on the number of trials N, using the following algorithm:

Algorithm:

1. Let w_1, w_2, w_3 be the 3 cube roots of unity. Define N random matrices $B_i = \{b_{jk}\}$ as follows:
   - $b_{jk} = 0$ if $a_{jk} = 0$
   - $b_{jk} \in w_1, w_2, w_3$ randomly, if $a_{jk} = 1$.
2. Set $X_i = det(B_i) * \overline{det(B_i)}$.
3. Estimate $Per(A)$ with $Y = \frac{X_1+...+X_N}{N}$.

The following two lemmas can be proved in a similar way to the proof given in section 2.6.1:

Lemma 2.9 *$E(X_i) = Per(A)$.*

Lemma 2.10 *$\frac{E(X_i^2)}{E(X_i)^2} \leq 2^{n/2}$.*

and as a consequence we get:

Theorem 2.8 *It is possible to get an (ϵ, δ)-approximation to $Per(A)$, if $N = 2^{n/2}\frac{36}{\epsilon^2}\log\frac{1}{\delta}$.*

3 Uniform Generation

The above section indicates the existence of a direct connection between counting problems and sampling algorithms. Jerrum, Valiant and Vazirani define and prove this connection in [JVV]. We will review their results here:

Let Σ be a finite alphabet, and R a relation which associates with each problem $x \in \Sigma^*$, a set of solutions $R(x) = \{y \in \Sigma^* | xRy\}$. Thus for example, x may be a formula in DNF, y an assignment to the variables of x, and xRy if y is a satisfying assignment of x. Given a problem instance x, we can define the following problems:

1. Existence - Does there exist a y, such that xRy ?
2. Construction - Find a y, such that xRy, if there exists one.
3. Uniform Generation - Generate uniformly at random a y, such that xRy.
4. Counting - What is the size of $R(x)$?

It is clear that the Existence problem is at least as easy as the Construction problem, in the sence that if we can construct a solution y, we can answer the corresponding Existence problem with 'yes', and if no y is constructed the answer is 'no'. It is also obvious that constructing some solution y is at least as easy as generating uniformly at random a solution y. Jerrum, Valiant and Vazirani show in their paper that the Generation problem is probably strictly more difficult the Construction problem, and easier then Counting. That is given an algorithm which finds $|R(x)|$ we ca build an algorithm which generates uniformly at random element of $R(x)$, but the converse is probably not true. Their results apply only for a special kind of relations, *self reducible relations*. Schnorr [S], was the first to study these type of relations. Self - reducible relations R, are relations for which there exists a simple inductive construction of solutions in $R(x)$ from the solutions of a few smaller instances of the problem x, and the length of each solution is polynomialy bounded by the length of x. A more formal definition follows:

Definition 3.1 A relation $R \subseteq \Sigma^* \times \Sigma^*$ is *self - reducible* iff -

1. There exists a polynomial p, such that $|y| = p(|x|)$ if xRy.
2. It can be tested deterministicly in polynomial time whether xRy.
3. There exist polynomial time computable functions $\psi : \Sigma^* \times \Sigma^* \to \Sigma^*$ and $\sigma : \Sigma^* \to N$ where
   (a) $\sigma(x) = O(log|x|)$.
   (b) If $p(x) > 0$ then $\sigma(x) > 0$.

(c) $|\psi(x,w)| \leq |x|$.

(d) $< x, y_1...y_n > \in R$ iff $< \psi(x, y_1...y_{\sigma(x)}), y_{\sigma(x)+1}...y_n > \in R$.

Now let us define Generating and Counting algorithms in the terms of relations R:

Definition 3.2 A *Uniform Generator* for a relation R is an algorithm which on input x, outputs only strings y in the solution set of x, each y with equal probability, and the probability for some output is bounded away from 0.

A *Deterministic Approximate Counter* for a relation R is an algorithm which on input $< x, \epsilon >$, outputs a Y such that

$$(1+\epsilon)^{-1}|R(x)| \leq Y \leq (1+\epsilon)|R(x)|$$

For self-reducible relations R, we can define the following set, of partial solutions of a problem instance x:

$$R_{x,w} = \{z | xRwz\}$$

The following lemma is easy to prove, using the self reducibility structure:

Lemma 3.1 *Let R be a self reducible relation. If there exists a deterministic approximate counter for R, then there exists a deterministic approximate counter for $R_{x,w}$ for any x.*

We now get to the main theorem of this section, which states that given a deterministic approximate counter for a self reducible R, we can construct a uniform generator for R. Note that the proof is constructive, and that the counting algorithm must be deterministic (for the connection between randomized approximate counters and generators, see section 4).

Theorem 3.1 *Let R be a self reducible relation. If there exists a deterministic approximate counter for R, then there exists a uniform generator for R.*

Proof: Let C be a deterministic approximate counter for R, and let $C_{x,y}$ be the approximation to $|R_{x,y}|$, as promised by the above lemma. We will describe a recursive algorithm which generates uniformly a $y = y_1 y_2 ... y_m \in R(x)$, $\quad m = p(|x|)$. Each iteration of the algorithm will generate the next letter y_i of y, with a probability proportional to the number of words in $R_{x,y_1 y_2 ... y_i}$. Set $y = \Lambda$ (the empty word). With probability $\frac{C_{x,y0}}{C_{x,y0}+C_{x,y1}}$ set the first letter of y to be 0, that is $y = y0$, or with probability $\frac{C_{x,y1}}{C_{x,y0}+C_{x,y1}}$ set the first letter to be 1. Continue recursively with the new y, until $|y| = m$. Then check if xRy, and if so output y with probability

$$\varphi = \frac{1}{C_{x,\lambda}} \frac{C_{x,0}+C_{x,1}}{C_{x,y_1}} \frac{C_{x,y_1 0}+C_{x,y_1 1}}{C_{x,y_1 y_2}} \cdots \frac{C_{x,y_1 y_2 ... y_{m-1} 0}+C_{x,y_1 y_2 ... y_{m-1} 1}}{C_{x,y_1 y_2 ... y_m}}$$

Note that if C was an exact counter, then $\varphi = \frac{1}{|R_{x,y_1 y_2 \cdots y_m}|} = 1$, that is we always output y, and each y is output with the same probability. Since C is only approximate, we output each y with probability ≤ 1, and the probability of each output y is the probability of generating it multiplied by φ, which is:

$$\frac{C_{x,y_1}}{C_{x,0}+C_{x,1}} \frac{C_{x,y_1 y_2}}{C_{x,y_1 0}+C_{x,y_1 1}} \cdots \frac{C_{x,y_1 y_2 \ldots y_m}}{C_{x,y_1 y_2 \ldots y_{m-1} 0}+C_{x,y_1 y_2 \ldots y_{m-1} 1}} \varphi = \frac{1}{C_x}$$

that is, each output y has an equal probability. The only thing which need to be checked is that the probability of some output is bounded away from 0, that is that φ is bounded away from 0. But $\varphi \geq (1+\epsilon)^{-2m}$, and for $\epsilon = \frac{1}{m}$ it is bounded away from 0. $\square$

Let us move on, and show that Uniform generation is probably more difficult then Construction. That is, there are relations R, for which there exist polynomial time bounded algorithms which construct solutions of R, but no fast uniform generator for R is believed to exist.

Theorem 3.2 *If there exists a polynomial time bounded uniform generator for cycles in a given directed graph, then $NP = RP$.*

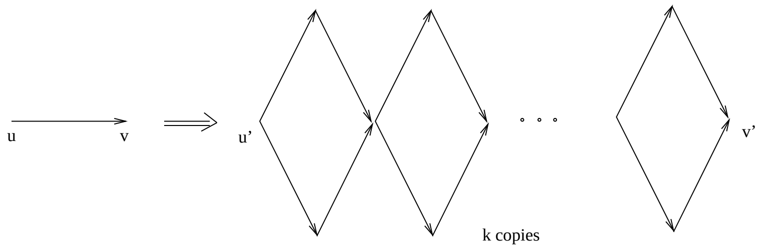
Proof: Let $G = (V, E)$ be a directed graph with $|V| = n$, and define

$$CYCLE(G) = \{c | c \textit{ is a cycle of } G\}$$

Then:

1. The problem of constructing a cycle in G is easily solved in polynomial time.
2. We will show that if there exists a uniform generator of cycles then $NP = RP$, and thus it is unlikely that uniform generation of cycles can be solved in polynomial time. Define
$$DHC = \{h | h \textit{ is a Hamiltonian cycle in } G\}$$

Let $G' = (V', E')$ be the directed graph derived from G by replacing every edge $< u, v > \in E$ by:



where $k = nlogn$.
We can see that for each cycle c of G, which includes edge $< u, v >$, there are 2^k cycles in G', which include nodes u', v'. Therefore, if G' contains a cycle of length $2kn$ then it contains at least 2^{kn} cycles of this length, and this happens iff G contains a Hamiltonian cycle (of length n). Also, the total number of cycles in G' of length $< 2kn$ is $< n^n 2^{k(n-1)} \leq 2^{kn}$. Thus if $|DHC| \neq 0$ then the probability that a randomly generated cycle of G' has length $2kn$ is $\geq 1/2$, and if $|DHC| = 0$ then the probability is 0. Therefore the problem of generating Hamiltonian cycles is in RP, and because it is NP - complete we get that $NP = RP$.

□

To illustrate the gap between Counting and Uniform Generating, Jerrum, Valiant and Vazirani turn to the problem of counting the number of satisfying assignments of a DNF formula versus the problem of generating uniformly a satisfying assignment to such a formula. Their proof relies on the Monte-Carlo algorithm presented in section 2.3, which is due to Karp and Luby.

Theorem 3.3 *There exists a polynomial time bounded uniform generator for satisfying assignments of a DNF formula, but the corresponding counting problem is #P-complete.*

Proof:

1. The problem of counting the number of satisfying assignments of a DNF formula was shown to be $\#P$-complete ([V]).

2. The sampling algorithm which is presented in section 2.3, can be used as a uniform generator. We generate uniformly at random a pair (j, x), where x is an assignment satisfying the j'th clause of the formula F, and output x if j is the lowest numbered clause satisfied by x (steps 1-3 of section 2.3). Each assignment is generated with equal probability, and the probability of some output is $\geq \frac{1}{m}$, where m is the number

of clauses of F. If we repeat this algorithm for m times, then the probability for no output will be $\leq (1 - 1/m)^m \leq 1/e$, and therefore bounded away from 1. $\square$

4 Almost Uniform Generation and Randomized Approximate Counting

In the last section we reviewed the connection between uniform generation and counting. We showed that given a counting algorithm (even an approximate one) we can construct a uniform generator, but that the opposite is probably not true. Jerrum, Valiant and Vazirani in [JVV], and Broder in [B] showed that if we are willing to compromise and be satisfied with a *randomized approximate counting* algorithm, then it is possible to construct one given an *almost uniform generator* and vice-versa. Their results apply only for *self reducible* relations.

Let us define the concepts above (these definitions can be found in [JVV],[B],[JS] and other places, in various forms):

Definition 4.1 An ϵ - *Generator* for a relation R is an algorithm, which on input x, outputs with probability bounded away from 0, only strings y in $R(x)$, and

$$\frac{(1+\epsilon)^{-1}}{|R(x)|} \leq Pr(output\ y) \leq \frac{(1+\epsilon)}{|R(x)|}$$

(that is each output y has a probability which is close to uniform).

An *Almost Uniform Generator* for R, is an algorithm which on input $< x, \epsilon >$, behaves as an ϵ - Generator for R.

An Almost Uniform Generator is *fully - polynomial* (f.p.) if it runs in time polynomial in $|x|$ and $\frac{1}{\epsilon}$.

An (ϵ, δ) *Randomized Approximate Counter* for a relation R, is an algorithm, which on input x, outputs a Y such that

$$Pr((1+\epsilon)^{-1}|R(x)| \leq Y \leq (1+\epsilon)|R(x)|) \geq 1-\delta$$

A *Randomized Approximate Counter* for R, is an algorithm, which on input $< x, \epsilon, \delta >$, behaves as an (ϵ, δ) Randomized Approximate Counter for R.

A Randomized Approximate Counter is *fully - polynomial* if it runs in time polynomial in $|x|$, $\frac{1}{\epsilon}$ and $\log \frac{1}{\delta}$.

We are now ready to state and prove the equivalence of randomized approximate counters and almost uniform generators for self reducible relations R. These results can again be found in [JVV] and [B].

Before we proceed, let us state the following lemma. The proof is omitted.

Lemma 4.1 *Let R be a self reducible relation. If there exists an f.p. randomized approximate counter for R, then there exists an f.p. randomized approximate counter for $R_{x,w}$.*

From this lemma, we get that given a fast approximate counter we can build a fast uniform generator. Again the proof is constructive.

Theorem 4.1 *Let R be a self- reducible relation. If there exists an f.p. randomized approximate counter for R, then there exists an f.p. almost uniform generator for R.*

Proof: The proof is identical to the one of theorem 3.1, with the difference that we are given a f.p. randomized approximate counter C instead of a deterministic one. Therefore the algorithm presented in theorem 3.1 constructs an almost uniform generator instead of a uniform one.

Let $m = p(|x|)$, then by the algorithm described in the proof of theorem 3.1, we make $2m$ calls to the randomized approximate counter C (using the above lemma). Therefore the deviation from the exact uniform distribution $\frac{1}{|R(x)|}$ is at most $2m\delta$. But $\frac{1}{|R(x)|} \geq \frac{1}{2^m}$, and thus if we take $\delta = \frac{\epsilon}{2m2^{m+1}}$ then:

If $< x, y > \in R$ then

$$\frac{(1+\epsilon)^{-1}}{|R(x)|} \leq \frac{1}{|R(x)|} - 2m\delta \leq Pr(output\ y) \leq \frac{1}{|R(x)|} + 2m\delta \leq \frac{(1+\epsilon)}{|R(x)|}$$

and if $< x, y > \notin R$ then $Pr(output\ y) = 0$.

It is easy to verify that the resulting uniform generator is polynomial in $|x|$ and $\frac{1}{\epsilon}$. $\square$

The converse of the above theorem is:

Theorem 4.2 *Let R be a self - reducible relation. If there exists an f.p. almost uniform generator for R, then there exists an f.p. randomized approximate counter for R.*

Proof: Let $m = p(|x|)$, and let M be an f.p. almost uniform generator for R. Make N calls to M with input $< x, \epsilon/m >$, and let w be the most commonly occurring prefix of length $\sigma(x)$ among the strings generated. Set $X_i = 1$ if the initial segment of the i'th string generated is w, and $X_i = 0$ otherwise. Take $N \geq \frac{|R(x)|m^2}{|R_{x,w}|\epsilon^2} \log \frac{2}{\delta}$, then $Y = \frac{X_1 + \ldots + X_N}{N}$ is an $(\frac{\epsilon}{m}, \delta)$-approximation to $\frac{|R_{x,w}|}{|R(x)|}$ (see section 2.2). But $\frac{|R_{x,w}|}{|R(x)|} \geq \frac{1}{2^{\sigma(x)}} \geq \frac{1}{|x|}$ (w was chosen as the most commonly occurring prefix of length $\sigma(x)$), and therefore N is polynomial in $|x|$, $\frac{1}{\epsilon}$ and $\log \frac{1}{\delta}$.

$R_{x,w} = R(\psi(x, w))$ because R is self- reducible, and thus we can proceed recursively. After at most m iterations, we will be left with a set containing just one string. Let Y' be the inverse of the product of all the ratios computed. Then Y' is an (ϵ, δ) approximation to $|R(x)|$, because

$$(1+\epsilon)^{-1}|R(x)| \leq (1+\frac{\epsilon}{m})^{-m}|R(x)| \leq Y' \leq (1+\frac{\epsilon}{m})^{m}|R(x)| \leq (1+\epsilon)|R(x)|$$

□

Theorem 4.3 *Let R be a self - reducible relation. If there exists a f.p. $|x|^k$ - generator for R, where the output probability of each $y \in R(x)$ is known (or can be computed in polynomial time), then there exists a f.p. randomized approximate counter for R.*

Proof: The proof is similar to the proof of theorem 4.2, and relies on the algorithm described in section 2.2 for computing the ratio of two sets. □

5 Markov Chains as Almost Uniform Generators

As stated in the above sections, there is a tight connection between counting and generating. Therefore it may be easier to try and build an almost uniform generator for a relation R, instead of trying to construct a counting algorithm for it. Broder was the first to use Markov Chains as generators. Before we review his results and see how to use Markov Chains as almost uniform generators, let us review some elementary concepts of Markov Chains. These definitions can be found in any probability book (see for example [Ro] for a good reference on Markov Chains).

5.1 Definitions and Notation

Let $(X_t)_{t=0}^{\infty}$ be a Markov-chain on a finite *space state* $[N] = \{1, ..., N\}$, with *transition matrix* $P = (p_{ij})_{i,j=1}^{N}$, where $p_{ij} = Pr(X_{t+1} = j | X_t = i)$, independent of t.

The *s-step transition matrix* is $P^s = (p_{ij}^s)$, where $p_{ij}^s = Pr(X_{t+s} = j | X_t = i)$, independent of t.

Let $\pi^t = (\pi_1^t, ..., \pi_N^t)$ be the *distribution vector* of X_t, where $\pi_i^t = Pr(X_t = i)$. The *initial distribution* of the chain is π^0. We will note $\pi_{ij}^t = Pr(X_t = i | X_0 = j)$, that is π_{ij}^t is the distribution vector of the chain at time t, if we are given that $\pi_j^0 = 1$ and $\pi_k^0 = 0$ for every $k \neq j$. In this case j is called the *initial state* of the chain.

The chain is *ergodic* if there exists a distribution $\pi = (\pi_1, ..., \pi_N)$, such that $\pi_i > 0$ for all i, and $\lim_{s \to \infty} p_{ij}^s = \pi_j$ for all $i, j \in [N]$. Then π is called the *stationary distribution* of the chain, and it is the only vector which satisfies $\pi P = \pi$, that is, it is a left eigenvector of P with eigenvalue $\lambda_0 = 1$.

The following lemma, which we state without proof, gives an easy criterion to verify whether a given chain is ergodic.

Lemma 5.1 *The chain is ergodic iff:*

1. *The chain is* irreducible *- for all $i, j \in [N]$, $\exists s$ such that $p_{ij}^s > 0$, and*
2. *The chain is* aperiodic *- for all $i, j \in [N]$, $\gcd\{s | p_{ij}^s > 0\} = 1$.*

An ergodic Markov chain is said to be *time - reversible* iff $p_{ij}\pi_i = p_{ji}\pi_j$ for every $i, j \in [N]$, where π is the stationary distribution of the chain.

If we can find a distribution π, such that $p_{ij}\pi_i = p_{ji}\pi_j$ for every i, j, then:

$$(\pi P)_j = \sum_i p_{ij}\pi_i = \sum_i p_{ji}\pi_j = \pi_j$$

for every j. Hence, the chain is reversible, and π is its stationary distribution.

Thus for example if $p_{ij} = \frac{1}{d}$ for every i and j, where d is some positive constant, then the uniform distribution $\pi_i = \frac{1}{|N|}$, is a solution to the set of equations $p_{ij}\pi_i = p_{ji}\pi_j$, and therefore the stationary distribution of the chain.

The *underlying graph* of a Markov chain is $G = (V, E)$, where $V = [N]$, and for each $i, j \in V$, such that $p_{ij} > 0$, there is a directed edge $(i, j) \in E$ with weight p_{ij}. State i will be called a *neighbor* of state j, if the edge (i, j) exists, that is if $p_{ij} > 0$. If also $p_{ji} > 0$, then states i, j will be called *neighbors*.

If $p_{ij} = \frac{1}{d_i}$, where d_i is the number of edges leaving state i, then as above, the distribution $\pi_i = \frac{d_i}{2|E|}$ is a solution to the set of equations $p_{ij}\pi_i = p_{ji}\pi_j$. Hence, the chain is reversible, with stationary distribution π.

5.2 Markov Chains as Generators

We now outline a general procedure, which describes how to use Markov chains as almost uniform generators.

Let R be a relation, with solution set $R(x)$ which we want to generate uniformly.

1. Define a Markov Chain on $R(x)$ with uniform (not necessarily) stationary distribution.
2. Simulate the chain until it is close to stationarity. That is, if we are at state i, then we move with probability p_{ij}, to state j.
3. Output the state reached as the output of an almost uniform generator.

The problems which this simple scheme raises, are many:

1. How do we define the Markov chain?
   This is not always easy, since the chain should be irreducible so that it will be possible to reach (generate) all states of the chain. The chain should also be aperiodic, so that at a given time there will be a positive probability to be at each of the states of the chain. But this requirement is easily achieved, if we add a self loop of probability $1/2$ to each state. That is we set $p_{ii} = 1/2$ for each i, and adjust the rest of the probabilities. This change will only slow slightly the simulation of the chain, but will result with an aperiodic chain.
2. From what state do we start simulating the chain?
   It should be possible to construct at least one element of $R(x)$ in polynomial time and to start the simulation from it.

3. When is the chain close to stationarity?
   For how long do we need to simulate the chain until it is close to stationarity, and what does it mean to be close to stationarity? (see section 6).

4. What is the stationary distribution of the chain?
   If the chain has uniform distribution, then it can be used as an almost uniform generator. However we will show that it suffices to have a chain with a distribution that is 'close' to uniform. This is also not always easy to achieve or determine.

The first two problems are directly connected to the structure of the problem x, and each problem should be treated differently. For example, Broder in [B] defined a Markov chain on the set of perfect and almost perfect matchings of a given graph. The problem of finding one perfect matching has many known algorithms. In section 9, we will define a Markov chain on the set of all independent sets of a given graph. Constructing one independent set is of course easy, just take any node of the graph as an independent set of size one. For this reason let us concentrate on the last two problems raised above. We will complete this section with a more concrete discussion of the last question and leave the third question for the next section.

If the stationary distribution is not uniform, but it is known that $\pi_i = \frac{c_i}{D}$, for $D = \sum_i c_i$, then we can output state i with probability $\frac{1}{c_i}$, and thus achieve a uniform distribution on $R(x)$. As a result we can build an approximate counter for R, whose running time will depened on the deviation of the stationary distribution from a uniform one. The resulting counting algorithm will be fully polynomial, if the stationary distribution is *close to uniform*, where:

Definition 5.1 The stationary distribution π of a Markov chain over state space $[N]$, is said to be *close to uniform* if $\frac{1}{cN} \leq \pi_i \leq \frac{c}{N}$ for every i, where c is polynomial in $\log N$ (or c is polynomial in $|x|$, where $R(x)$ is the relation described by the Markov chain) .

(See also section 2.2 and theorem 4.3).

If the chain is reversible, it is possible to compute the constants c_i, as the simulation proceeds.

Algorithm:

1. Assume without loss of generality that the simulation of the chain begins at state i. Set $c_i = 1$.

2. The chain is reversible and therefore

$$p_{ij}\pi_i = p_{ji}\pi_j$$

hence also

$$p_{ij}c_i = p_{ji}c_j$$

Therefore, for each j such that $p_{ij} > 0$, we can compute

$$c_j = \frac{p_{ij}}{p_{ji}} c_i$$

Since we can move from state i only to states j for which $p_{ij} > 0$, we are done. Continue with state j.

If the chain is not reversible, but we know that $\frac{\pi_i}{\pi_j} \leq c$ for any two neighbor states i, j and c a constant, then we can derive the bound $\frac{\pi_i}{\pi_j} \leq c^d$, for any two states i, j, where d is the diameter of the underlying graph of the chain. Thus if $d \leq \log\log N$, where N is the number of states, (or $d \leq \log|x|$ in terms of the relation $R(x)$ described by the chain) then $\frac{\pi_i}{\pi_j} \leq (\log N)^c$, hence the stationary distribution π, is close to uniform.

6 The Rate of Convergence of Markov Chains

Let us turn now to the question of estimating the time it takes a given Markov chain MC, to get close to its stationary distribution π, or in a more formal way: to estimate the rate of convergence of MC. Fast rate of convergence is needed if we want to use the chain as a polynomial generator. It is not enough to know that the chain converges fast. We need to know also how fast, so that we know for how long we should simulate the chain until it is close to stationarity. Before we introduce methods for estimating the rate of convergence of a given Markov chain, let us define more formally what we mean by 'close to stationarity' and 'converges fast':

Definition 6.1 Let d(t) be some distance measure between the stationary distribution π and the distribution π^t of the chain at time t. We say the chain is $\epsilon - close$ to π at time t if $d(t) \leq \epsilon$.

Definition 6.2 A Markov Chain over state space $[N]$, *converges rapidly* to its stationary distribution π, if it is $\epsilon - close$ to π at time t, where t is a polynomial in $\log N$ and $\frac{1}{\epsilon}$.

We will now introduce two methods to estimate the time t needed, until the chain is $\epsilon - close$ to its stationary distribution - one developed by Sinclair and Jerrum [SJ], and the other by Aldous [A].

6.1 Conductance

This method for estimating the rate of convergence of a given Markov chain is due to Jerrum and Sinclair [SJ]. The intuition behind their method is that the chain converges rapidly if it is unlikely to stay in any subset S, of the state space, whose total stationary distribution is small. They formalize this concept by considering the underlying graph of the chain as a network, and showing that the edges which come out of S, should carry a large 'flow', so that S is easyly left. The conductance of the chain measures the minimum relative connection between small subsets S and the rest of the space, and is shown to be related to the rate of convergence of the chain. We outline now the main aspects of their proof, but first let us define the distance measure they use:

Definition 6.3 The *relative pointwise distance* between π and π^t is

$$\Delta(t) = \max_{i,j \in [N]} \frac{|\pi^t_{ij} - \pi_i|}{\pi_i}$$

The proof of the following theorem can be found in [SJ]:

Theorem 6.1 *Let P be the transition matrix of an ergodic time-reversible Markov chain, let λ_i, $1 \leq i \leq N$, be the eigenvalues of P, and assume that $\lambda_1 = 1 \geq \lambda_2 \geq \ldots \geq \lambda_N$. Then*

$$\Delta(t) \leq \frac{\lambda_2^t}{\min_{i \in [N]} \pi_i}$$

As a result of this theorem, we see that an ergodic time-reversible Markov chain will converge rapidly to π, if π_i is not too small for all $i \in [N]$, and λ_2 is bounded away from 1. The first of these conditions can be easily checked if the stationary distribution is uniform or known to be close to uniform, therefore we now examine the second.

Let $S \subset [N]$ and define -

$C_S = \sum_{i \in S} \pi_i$ - the *capacity* of S,

$F_S = \sum_{i \in S, j \in \bar{S}} p_{ij} \pi_i$ - the *flow* of S,

and $\Phi_S = F_S / C_S$.

Let $\Phi = \min_{0 < |S| < N, C_S \leq \frac{1}{2}} \Phi_S$ be the *global conductance* of the chain.

If the chain is time-reversible then $F_S = F_{\bar{S}}$ and therefore

$$\Phi = \min_{0 < |S| < N} \max\{\Phi_S, \Phi_{\bar{S}}\}$$

Jerrum and Sinclair show now the connection between the second eigenvalue λ_2 and the conductance of the chain:

Lemma 6.1 *For an ergodic time- reversible Markov chain, $\lambda_2 \leq 1 - \Phi^2/2$.*

As a result they get the following criterion for the rate of convergence of the chain in terms of the conductance.

Theorem 6.2 *For any ergodic time-reversible Markov chain*

$$\Delta(t) \leq \frac{(1 - \Phi^2/2)^t}{\min_{i \in [N]} \pi_i}$$

From here we get:

Corollary 6.1 *An ergodic time-reversible Markov chain will be $\epsilon -$ close to its stationary distribution after $t = O(\Phi^{-2} \log(\frac{1}{\pi_{min}}) \frac{1}{\epsilon})$ simulation steps. The chain will converge rapidly if Φ is polynomial in $\log |N|$, and π is close to uniform.*

We replaced the problem of estimating the rate of convergence of the chain, with that of estimating the conductance Φ. So how do we estimate Φ? Jerrum and Sinclair in [JS], use the conductance technique to estimate the rate of convergence of a Markov chain defined on the set of matchings of a bipartite graph (see section 8). They defined a unique, *canonical* path between any two states of the chain, and showed how the maximal number of canonical paths traversing any edge of the underlying graph of the chain, relates to the conductance of the chain. We describe this connection between canonical paths and the conductance of any ergodic time-reversible Markov chain in the following lemma:

Lemma 6.2 *Let MC be an ergodic time-reversible Markov chain with uniform stationary distribution π and space state $[N]$. Assume all transition probabilities p_{ij} are either 0 or $\geq p$. Let there be defined a canonical path between any two states of the chain, and assume that the maximum number of canonical paths traversing any edge of the underlying graph of the chain is $\leq b$. Then the global conductance of the chain $\Phi \geq \frac{p|N|}{2b}$.*

Proof: By definition of the global conductance we have -

$$\Phi = \min_{0<|S|\leq|N|/2} \frac{\sum_{i\in S, j\notin S} p_{ij}\pi_i}{\sum_{i\in S}\pi_i} \geq \min_{0<|S|\leq|N|/2} \frac{\sum_{i\in S, j\notin S} p\pi_i}{\sum_{i\in S}\pi_i} = p \min_{0<|S|\leq|N|/2} \frac{|C(S)|}{|S|}$$

where $C(S)$ is the cut defined by S. Notice that the sum above is only over states $i \in S$, $j \notin S$, for which $p_{ij} > 0$ (which define exactly the edges in the cut $C(S)$). But since the chain is ergodic, and therefore irreducible, there must exist for every set S, states $i \in S$, $j \notin S$ such that $p_{ij} > 0$.

For any set S, the number of paths which cross from S to $N - S$ is $|S|(|N| - |S|) \geq \frac{|S||N|}{2}$. Hence if b is the maximum number of canonical paths traversing any edge, then

$$|C(S)| \geq \frac{|S||N|}{2b}$$

and therefore

$$\Phi \geq \frac{p|N|}{2b}$$

□

Recently [MV] proved a similar result to the one stated in theorem 6.2, for general Markov chains. The distance measure they used was:

Definition 6.4 *The normalized variation distance* between π and π^t is

$$d_n(t) = \max_j \sum_{i\in[N]} \frac{(\pi^t_{ij} - \pi_i)^2}{\pi_i}$$

Their result applies only for *strongly aperiodic* Markov chains, that is, $p_{ii} \geq 1/2$ for each state i.

Theorem 6.3 *For any ergodic and strongly aperiodic Markov chain and for any initial distribution π^0, it is true that $d_n(t) \leq (1 - \Phi^2)^t d_n(0)$.*

6.2 Coupling

The Coupling technique is due to Aldous [A]. The main idea behind it, is to start the simulation of the given Markov chain at two different states and see when the two processes meet. Each of the two "copies" of the chain behaves as the original chain, but their behavior as a pair may be coordinated. Now, suppose one process is started at the stationary distribution and the other at any fixed distribution. Then by the time the two processes meet, the second will also be at the stationary distribution, and this time certainly bounds the rate of convergence of the chain. The time it takes the processes to meet is called their coupling time. The formall definition of Coupling is:

Let X_t be a Markov process on the set of states N. A *Coupling* is a pair of processes (X_t^j, X_t^k), such that:

- X_t^j is distributed as X, given that $X_0 = j$.
- X_t^k is distributed as X, given that $X_0 = k$.
- $X_t^j = X_t^k$ for $t > T_{jk}$, where $T_{jk} = inf\{t | X_t^j = X_t^k\}$ (that is, once the processes meet, they continue together).

The *Coupling Time* of the chain is:

$$T = \max_{j,k \in [N]} T_{jk}$$

As we stated above, there is a tight connection between the coupling time and the rate of convergence of the chain. Before we describe this connection, let us define the distance measure used:

Definition 6.5 The *Variation Distance* between π and π^t is

$$d(t) = \max_{j \in [N]} \frac{1}{2} \sum_{i \in [N]} |\pi_{ij}^t - \pi_i|$$

The following lemma, can be found in [A]:

Lemma 6.3 *For any coupling of a Markov process X_t, $d(t) \leq Pr(T > t)$.*

Proof: Let X_t^j and X_t^k be a coupling of X_t. Then, by the triangle inequality we get that-

$$\begin{aligned} d(t) &\leq \max_{j,k\in[N]} \sum_{i\in[N]} |\pi_{ij}^t - \pi_{ik}^t| \\ &\leq \max_{j,k\in[N]} Pr(X_t^j \neq X_t^k | X_0^1 = j, X_0^2 = k) \\ &= \max_{j,k\in[N]} Pr(T_{jk} > t) = Pr(T > t) \end{aligned}$$

□

The bound given in the lemma is tight, that is for any Markov process there exists a coupling such that $d(t) = Pr(T > t)$ (see [P] or [G]).

Theorem 6.4 $d(E(T)\frac{1}{\epsilon}) \leq \epsilon$.

Proof: Follows from lemma 6.3 and the Markov inequality. □

As result we get:

Corollary 6.2 *An ergodic Markov Chain is $\epsilon -$ close to its stationary distribution, after time which is polynomial in the expected coupling time of the chain and in $\frac{1}{\epsilon}$.*

Corollary 6.3 *An ergodic Markov chain over state space $[N]$, is rapidly converging, if its expected coupling time is polynomial in $\log \frac{1}{N}$.*

So what we need, is to try and bound the expected coupling time of a given chain. We first have to prove an important lemma, which is basicly a version of the Gambler's ruin problem. We shall use this lemma later, to bound the expected coupling time.

Lemma 6.4 *Consider the simple random walk on a line, with states $\{0, 1, ..., n\}$, probability $p > 0$ of moving to the left, q of moving to the right, and r of remaining in the present state, and such that 0 is an absorbing state and n is partially reflecting (that is probability p of moving from n to $n-1$, and $1-p$ of staying at n). Then the expected time until 0 is reached is*

- $\frac{n}{p}$ *for* $q = 0$.
- $O(\frac{n^2}{2p})$ *for* $p = q$
- $O(\min\{\frac{n}{p-q}, \frac{n^2}{2p}\})$ *for* $p > q$.
- $O(\frac{n}{q-p}(\frac{q}{p})^n)$ *for* $p < q$.

Proof: Let $E(x)$ be the expected time to reach 0 when starting at x. Then we have the recursive set of equations:

$$E(x) = pE(x-1) + qE(x+1) + rE(x) + 1 \qquad 0 < x < n$$

$$E(0) = 0 \quad , \quad E(n) = pE(n-1) + (q+r)E(n)$$

If we solve them we get that:

$$E(n) = \frac{n}{p} + \frac{n}{p}\sum_{i=1}^{n-1}(\frac{q}{p})^i - \frac{1}{p}\sum_{i=1}^{n-1} i(\frac{q}{p})^i$$

Then for -

- $q = 0$, $E(n) = \frac{n}{p}$.
- $q = p$, $E(n) \leq \frac{n^2}{2p}$.
- $p > q$, $E(n) \leq \min\{\frac{n}{p-q}, \frac{n^2}{2p}\}$.
- $p < q$, $E(n) \leq \frac{n}{q-p}(\frac{q}{p})^n$.

□

We proved the lemma for a random walk which takes one step to the right or the left. The lemma will also be true for a random walk which advances in jumps, if the size of the jump to the left equals the size of the jump to the right. We shall not prove this generalization here, because we need only the simple case of a step advance.

As a consequence of the lemma we get:

Theorem 6.5 *Let (X_t^1, X_t^2) be a coupling of a Markov process X_t on a set of states N. Assume there is an integral distance d_N, defined on N, and for every pair of states $i, j \in N$, $d_N(i,j) < n$. If*

$$|d_N(X_t^1, X_t^2) - d_N(X_{t+1}^1, X_{t+1}^2)| \leq 1$$

for every t, and

$$p = Pr[d_N(X_t^1, X_t^2) < d_N(X_{t+1}^1, X_{t+1}^2)] \leq Pr[d_N(X_t^1, X_t^2) > d_N(X_{t+1}^1, X_{t+1}^2)] = q$$

then the expected coupling time is polynomial in n, p and q.

Proof: By the definition of the coupling time, $T = inf\{t|d_N(Z_t^1, Z_t^2) = 0\}$. We can view the coupling as a random walk on the line $\{0, 1, ..., n\}$ with an absorbing barrier at 0, and a partially reflecting one at n, and such that at each step there is probability p to move to the left, and q to the right and $p \geq q$. The coupling time will be the time when the walk is absorbed in 0. Thus by the above lemma the expected coupling time is $O(\min\{\frac{n^2}{2p}, \frac{n}{p-q}\})$. $\square$

Up to now all the examples found in the literature, which used Coupling to prove rapid convergence, had a positive probability p, to reduce the coupling distance at each step, and the probability q to increase the distance, was 0. We will show in the next sections examples of random walks with $p > 0$ and $q > 0$.

7 Random Walks on the n-Dimensional Cube

Let us now move to some concrete examples of rapidly converging Markov chains. We will start with two examples of random walks on the n-dimensional cube, the first over Z_2 and the second over Z_k. The first walk is a private example of the second, but since it illustrates in a simple way the conductance technique, we bring it here. The walk can also be proved to converge rapidly by coupling arguments (see [A]). The second walk will be proved to converge rapidly using the coupling technique, and as far as we know does not have a different proof of rapid convergence.

7.1 The n-dimensional Cube over Z_2

Let N be the set of all vectors $(x_1, ..., x_n)$ over $Z_2 = \{0, 1\}$.

Define a Markov Chain on N as follows:
Given a vector u, choose one coordinate uniformly at random, and with probability 1/2 change it.

The underlying graph of this chain is the n - dimensional cube C_{n2}.

The following lemma bounds the conductance of the chain. The proof relays on the proof given by Jerrum and Sinclair in [JS], which bounds the conductance of a Markov chain defined on the set of perfect matchings of a bipartite dense graph. We shall review their proof briefly in section 8.

Lemma 7.1 *Let Φ be the global conductance of C_{n2}. Then $\Phi \geq \frac{1}{4n}$.*

Proof: By the definition of the Markov Chain, we have $p_{ij} \geq \frac{1}{2n} \stackrel{\text{def}}{=} p$ for each two adjacent vertices i, j, and $p_{ij} = 0$ otherwise. The chain is clearly irreducible, and also aperiodic since $p_{ii} = \frac{1}{2}$ for each state i. Hence the chain is ergodic. It is also easy to check that it is time-reversible (see section 5), with uniform stationary distribution. Thus by lemma 6.2

$$\Phi \geq \frac{p|N|}{2b} = \frac{|N|}{4b}$$

where b is the maximal number of canonical paths traversing any edge of the cube.

We will define a canonical path between any pair of vectors $u, v \in N$, and show that $b \leq |N|$, and therefore $\Phi \geq \frac{1}{4n}$.

The *canonical path* between u and v will consist of scanning u from left to right and changing every coordinate which is different from v.

Let t be an arbitrary edge of C_{n2} corresponding to a transition from m to m', and let $P(t) = \{(u, v) | \textit{there is a canonical path between } u \textit{ and } v \textit{ traversing } t\}$. Define an injective mapping σ_t from $P(t)$ to N by: $\sigma_t(u,v) = u \oplus v$. It is clear that $\sigma_t(u,v) \in N$, and given $\sigma_t(u,v)$ it is easy to reconstruct u and v in the following way -

- Each coordinate of $\sigma_t(u,v)$ which is 0, is in $u \cap v$ and can be found in m.
- The rest of the coordinates can be reconstructed from m: Each coordinate which is to the left of the coordinate changed by the transition m, m', has already been changed and therefore belongs to v. Each coordinate to the right of the coordinate changed by transition m, m', including it, belongs to u.

Therefore $|P(t)| \leq |N|$, and hence $b \leq |N|$. $\square$

Corollary 7.1 *The random walk on C_{n2} is $\epsilon -$ close to a uniform stationary distribution after $16n^3\frac{1}{\epsilon}$ steps.*

Note: It is easy to see, using the Coupon Collector lemma, that the real rate of convergence in this case is much faster, $O(n \log n)$. See Aldous [A1], for a full analyses.

7.2 The n-dimensional Cube over Z_k

We will now examine the random walk on the n-dimensional cube over Z_k. We shall prove it converges rapidly to a uniform stationary distribution, using coupling arguments. As far as we know, rapid convergence of this walk was not proved elsewhere.

Let N be the set of all vectors $(x_1, ..., x_n)$ over $Z_k = \{0, 1, ..., k-1\}$. Define a distance on N, for every $U, V \in N$:

$$d(U,V) = \sum_{i=1}^{n} \delta_k(U[i], V[i])$$

where

$$\delta_k(a,b) = \min\{b-a, a+k-b\} \qquad assuming \quad a < b$$

For each $U \in N$, define the vectors U_i^- and U_i^+:

- $U_i^-[i] = (U[i]-1) \bmod k$, and $U_i^-[j] = U[j]$ for $j \neq i$.
- $U_i^+[i] = (U[i]+1) \bmod k$, and $U_i^+[j] = U[j]$ for $j \neq i$.

Define a Markov Chain $MC(N)$ on N, as follows:

Given a vector U, choose a coordinate i uniformly at random, and with probability 1/4 move to U_i^-, with probability 1/4 to U_i^+, and with probability 1/2 stay at U.

The underlying graph of this chain is the n - dimensional cube C_{nk}.

The Markov Chain defined by this walk is clearly irreducible and aperiodic, hence ergodic. The stationary distribution can be shown to be uniform (see section 5).

A **coupling** of this process is:

1. Let X_t^1 and X_t^2 be two dependent processes, behaving as defined by the following random walk. Assume they are in vertices U and V of C_{nk} respectively. Both processes choose uniformly at random the same coordinate i.
2. If $U[i] = V[i]$, then both processes move together, as defined by $MC(N)$.
3. If $U[i] \neq V[i]$, then with equal probability -
   - X_t^1 moves to either U_i^- or to U_i^+ and X_t^2 remains at V, or
   - X_t^2 moves to either V_i^- or to V_i^+ and X_t^1 remains at U.

It is easy to verify that this is indeed a coupling, that is:

- Each one of the processes is distributed as the original process.
- Once the processes meet, they continue together.

Lemma 7.2 *In each step of the coupling above there is probability $p \geq \frac{1}{2n}$ to reduce the distance between the two processes, and probability $q \leq p$ to increase it.*

Proof: Assume X_t^1 and X_t^2 are in vertices U and V of C_{nk} respectively, and that $d(U, V) = d$. The distance d can only be changed if the two processes choose a coordinate i, such that $U[i] \neq V[i]$. With out loss of generality assume that $U[i] > V[i]$. Then

- If X_t^1 moves to U_i^- and X_t^2 remains at V, or X_t^2 moves to V_i^+ and X_t^1 remains at U, then the distance changes to $d - 1$.
- If X_t^1 moves to U_i^+ and X_t^2 remains at V, then
  - If $U[i] < k - 1$ then the distance is $d + 1$.
  - If $U[i] = k - 1$ and $V[i] < k/2$ then the distance changes to $d - 1$.
  - If $U[i] = k - 1$ and $V[i] = k/2$ then the distance remains d.
  - If $U[i] = k - 1$ and $V[i] > k/2$ then the distance changes to $d + 1$.

and in a similar way, if X_t^2 moves to V_i^- and X_t^1 remains at U.

Thus in at least half of the possible transitions the distance is reduced, and in at most half of the possible transitions the distance is increased, hence $p \geq q$, and since there is at least one coordinate such that $U[i] \neq V[i]$, then $p \geq \frac{1}{2n}$. $\Box$

As a result:

Theorem 7.1 *The expected coupling time is* $\leq \frac{n^3 k^2}{4}$.

Proof: For any two vectors $U, V \in N$ we have that $d(U,V) \leq \frac{kn}{2}$. Therefore by theorem 6.5, we get that the expected coupling time is $\leq \frac{n^2 k^2}{4 \cdot 2p}$, where p is the probability to reduce the distance between U and V in each step of the coupling. From the above lemma $p \geq \frac{1}{2n}$, and thus the expected coupling time is $\leq \frac{n^3 k^2}{4}$. $\Box$

Corollary 7.2 *The random walk on* C_{nk} *is* $\epsilon -$ *close to a uniform stationary distribution after* $\frac{n^3 k^2}{4} \frac{1}{\epsilon}$ *steps.*

8 Approximating the Permanent for Dense Graphs

In section 2.6 we reviewed a Monte-Carlo algorithm which computes the Permanent, or an equivalent problem, the number of perfect matching of a bipartite graph. The complexity of the algorithm that was introduced there, was $O(2^{n/2})$, which is better then the fastest known deterministic algorithm, but still exponential. We will now show a randomized approximation algorithm which finds the number of perfect matchings of a bipartite graph $G = (V_1, V_2, E)$ in polynomial time, providing the graph G is *dense*, i.e the minimum degree of each edge is at least $n/2$, where $n = |V_1| = |V_2|$. As we stated in section 5, Broder was the first to use Markov chains as almost uniform generators ([B]). He defined a Markov chain on the set of perfect and almost perfect matchings of G, and showed that if the chain converges rapidly to its stationary distribution, then it can be used as an f.p. almost uniform generator of perfect matchings, and thus an f.p. randomized counter for perfect matchings exists. Jerrum and Sinclair in [JS], were the first to prove that the chain Broder defined, indeed converges rapidly. For the sake of completeness, we shall review their proof briefly:

Let $M_k(G)$ be the set of matchings in G of size k, and let $MC(G)$ be a Markov Chain with state space $N = M_n(G) \bigcup M_{n-1}(G)$ and transitions defined as follows :

Given a state $M \in N$, choose an edge $e = (i, j) \in E$ uniformly at random, and with probability 1/2 either stay at M or move to one of the following states -

1. if $M \in M_n(G)$ and $e \in M$, move to $M - e$.
2. if $M \in M_{n-1}(G)$ and $e \notin M$, move to $M + e$.
3. if $M \in M_{n-1}(G)$, i is not matched in M and $(j, k) \in M$ for some l, move to $M+e-(j, k)$.

Lemma 8.1 *Let $MC(G)$ be defined as above, and let Φ be the global conductance of the chain. Then $\Phi \geq \frac{1}{12n^6}$.*

Proof: By the definition of $MC(G)$ we have $p_{ij} \geq \frac{1}{2|E|} \geq \frac{1}{2n^2} \stackrel{\text{def}}{=} p$ for each pair of states i, j which are adjacent in the underlying graph and $p_{ij} = 0$ otherwise. It is easy to verify that $MC(G)$ is irreducible, and it is clearly aperiodic, hence ergodic. The chain is also time-reversible with uniform stationary distribution (see section 5). Thus by lemma 6.2

$$\Phi \geq \frac{p|N|}{2b}$$

where b is the maximal number of canonical paths which traverse any edge of the underlying graph of $MC(G)$.

[JS] show that $b \leq 3n^4|N|$, and since $p = \frac{1}{2n^2}$, the desired bound on Φ is achieved. The crux of their proof shows that there exists an injective mapping from the set of ordered pairs of perfect matchings (I, F) such that there is a canonical path from I to F which traverses a certain edge of the underlying graph, to the set N. □

Theorem 8.1 *There exists a f.p. randomized approximate counter for the permanent of dense matrices.*

Proof: By theorem 6.2 and the above lemma, we have that $MC(G)$ converges rapidly. Therefore we have an f.p. almost uniform generator for elements of N, and thus by [B] an f.p. randomized approximate counter for $|M_n(G)|$. □

9 Independent Sets in Graphs

Let $G = (V, E)$ be a graph, $|V| = n$. An *independent set* of G is a subset W of V, such that no two vertices W are adjacent in G. A *maximum independent set* is an independent set W, such that for every other independent set U of G, $|U| \leq |W|$.

We would like to find the number of all independent sets of G. This problem was shown to be $\#P$-complete [PB].

Theorem 9.1 *Counting matchings is not harder then counting independent sets.*

Proof: Let $G = (V, E)$ a graph and let $G' = (V', E')$ be its line graph. Then there is a one to one correspondence between matchings in G and independent sets in G', and furthermore between maximum matchings in G and maximum independent sets in G'. $\square$

Dagum showed recently in [D], that for claw free graphs G, i.e., graphs which do not contain an induced subgraph isomorphic to $K_{1,3}$, it is possible to construct an f.p. (ϵ, δ) approximation to the number of independent sets of G, using technics similar to the ones in [JS] to bound the conductance of the *exchange graph* of G. The problem of whether there exists an f.p. randomized approximate counter for the number of independent sets of a general graph G, remains open.

We will try to use coupling technics to show rapid convergence of Markov chains defined on the set of independent sets of G. Before we proceed, let us introduce the notation that will be used throughout this section.

Notation:
Let $N(G)$ be the set of all independent sets of G, and let $N = |N(G)|$. Assume G is connected and define an order on its vertices. For each vertex $v \in V$ and each subset $A \subset V$ define $\Gamma_A(v) = \{a | (a, v) \in E \quad , \quad a \in A\}$, the set of vertices in A which are adjacent to v. Let $|A \oplus B|$ denote the symmetric difference of A and B.

We can now start, and the most natural way is to try and define a symmetric walk $MC(G)$, on the set of independent sets of G, where two independent sets are neighbors if their symmetric difference is one. If we can show that such a walk exists and converges rapidly, then we will have almost immediately an f.p. randomized approximate counter for the independent sets of G. A formal description of $MC(G)$ follows:

1. Given a set $M \in N(G)$, choose a vertex $v \in V$ uniformly at random.

2. If $v \in M$ then with probability $1/2$, move to $M - v$.

3. If $v \notin M$ and $M + v \in N(G)$, then with probability $1/2$, move to $M + v$.

4. Else stay at M.

The following lemma proves that the above random walk is irreducible:

Lemma 9.1 *Let $A, B \in N(G)$ such that $|A \oplus B| = k$. Then there is path of length k between A and B in the underlying graph of $MC(G)$.*

Proof: Start with A. Delete all vertices in $A - B$ and then add all vertices in $B - A$. $\square$

Lemma 9.2 *The Markov chain described by $MC(G)$ is ergodic.*

Proof: The chain is irreducible by the above lemma, and it is clearly aperiodic since there is a probability of $1/2$ to stay in each given state. $\square$

Also, $p_{AB} = p_{BA} = \frac{1}{2n}$ for any two neighbor sets $A, B \in N(G)$, hence the walk is reversible with uniform stationary distribution (see section 5). However we can prove it converges rapidly only for the simple case of graphs G with degree ≤ 2. For such graphs, it is possible of course to count exactly the number of independent sets in polynomial time, but the proof of rapid convergence of $MC(G)$ uses arguments which may be used later for more general graphs, and therefore is worthwhile examining.

9.1 Independent Sets of Cycles and Lines

For the simple graph G with degrees ≤ 2, that is a graph which is a line or a cycle, it is true that the symmetric random walk $MC(G)$, converges rapidly. Let us define the walk for such graphs in a slightly different way:

1. Given a state $M \in N(G)$, choose a vertex $v \in V$ uniformly at random.
2. If $v \in M$ has degree 1, then with probability $1/4$, delete v .
3. If $v \notin M$ then
   - If $|\Gamma_M(v)| = 0$, then with probability $1/2$, add v.
   - If $|\Gamma_M(v)| = 1$, then with probability $1/4$, delete $\Gamma_M(v)$.
   - If $|\Gamma_M(v)| = 2$, then with probability $1/2$, delete either one of $v's$ neighbors, each one with equal probability.

Again, it is east to verify from the definition of the random walk that:

- $p_{AB} = \frac{1}{2n}$ if $|A \oplus B| = 1$.
- $p_{AB} = 0$ if $|A \oplus B| > 1$.

Hence the walk is reversible with uniform stationary distribution. We now show that it converges rapidly, using Coupling arguments.

A **Coupling** of this random walk:

1. Let X_t^1 and X_t^2 be two dependent processes behaving as defined by the following random walk. Assume they are in states A and B respectively, and that $|A \oplus B| = k$. Both processes choose uniformly at random the same $v \in V$.
2. If $v \in A$, $v \in B$ and v has degree one ,then with probability $1/4$, X_t^1 moves to $A - v$ and X_t^2 moves to $B - v$ (Clearly $|A \oplus B| = k$).
3. If $v \notin A$ and $v \notin B$ then -
   - If $\Gamma_A(v) = \Gamma_B(v)$ then both processes move together, as defined by the symmetric random walk above (again $|A \oplus B| = k$).
   - If $\Gamma_A(v) \neq \Gamma_B(v)$ then
     - If $|\Gamma_A(v)| = 0$, and $|\Gamma_B(v)| = 2$, then with probability $1/2$, X_t^1 moves to $A + v$ and X_t^2 deletes either one of $v's$ neighbors ($|A \oplus B| = k$).
     - If $|\Gamma_A(v)| = 1$, and $|\Gamma_B(v)| = 2$, then with probability $1/4$ both processes delete $\Gamma_A(v)$, (notice $\Gamma_A(v) \subset \Gamma_B(v)$), and with probability $1/4$ X_t^2 removes $\Gamma_B(v) - \Gamma_A(v)$ and X_t^1 remains at A. (Then with probability $1/4$, $|A \oplus B| = k - 1$ and with probability $3/4$, $|A \oplus B| = k$).
     - If $|\Gamma_A(v)| = |\Gamma_B(v)| = 1$, (and $\Gamma_A(v) \neq \Gamma_B(v)$), then with probability $1/4$, X_t^1 deletes $\Gamma_A(v)$ and X_t^2 deletes $\Gamma_B(v)$. (With probability $3/4$, $|A \oplus B| = k$ and with probability $1/4$, $|A \oplus B| = k - 2$).
     - If $|\Gamma_A(v)| = 0$, and $|\Gamma_B(v)| = 1$, then with probability $1/2$, X_t^1 moves to $A + v$ and with probability $1/4$, X_t^2 deletes $\Gamma_B(v)$. In this case with probability $3/4$, $|A \oplus B| = k$ and with probability $1/4$, $|A \oplus B| = k + 1$.
4. If $v \notin A$ but $v \in B$, then if v has degree 2 then B does nothing and A as described by the symmetric random walk above. If v has degree 1, then with probability $1/4$, both processes do nothing, with probability $1/4$, X_t^2 deletes v and X_t^1 does nothing, and with probability $1/2$, X_t^2 does nothing and X_t^1 -
   - Adds v if $|\Gamma_A(v)| = 0$
   - Deletes, with probability $1/2$, $\Gamma_A(v)$, if $|\Gamma_A(v)| = 1$.
   - Deletes with equal probability a vertex from $\Gamma_A(v)$, if $|\Gamma_A(v)| = 2$.

(Then with probability $\geq 1/4$, $|A \oplus B| = k-1$, and with probability $\leq 3/4$, $|A \oplus B| = k$).

It is easy to check, that the above description is indeed a coupling, that is both processes are distributed as the symmetric random walk $MC(G)$, and after they meet they continue together. The coupling time is showed to be polynomial by the following lemma:

Theorem 9.2 *The expected coupling time is* $\leq 2n^3$.

Proof: We will show that in each step of the coupling the probability q to increase the distance between the two processes is $\leq$ then the probability p to reduce that distance, and that $p \geq \frac{1}{4n}$. Let $A, B \in N(G)$. For each $u \in B - A$ define a set

$$S_u = \{v | v \notin A, B \;\; |\Gamma_A(v)| = 0 \;\; and \; \Gamma_B(v) = u\}$$

and similary for $u \in A - B$. Then for any vertex u, $|S_u| \leq 2$ (G has degree ≤ 2). If $|S_u| = 2$, and we choose u in the coupling process, then with probability $1/2$ the distance will be reduced by one. If we choose $v \in S_u$ then with probability $1/4$ the distance will increase by one. If $|S_u| = 1$ and we choose u, then with probability $\geq 1/4$ the distance is reduced by one. If we choose $v = S_u$ then with probability $1/4$ the distance is increased by one. Thus overall, for each set of vertices $\{u, S_u\}$, the probability to reduce the distance equals the probability to increase it. For vertices v, for which $v \notin S_u$ for any u, it is not possible to increase the distance. Hence $p \geq q$. Also by the above arguments, if we choose a vertex $u \in |A \oplus B|$, then we have a probability $\geq 1/4$ to reduce the distance, and thus $p \geq \frac{|A \oplus B|}{4n} \geq \frac{1}{4n}$. Therefore by theorem 6.5, the expected coupling time is $\leq 2n^3$. $\square$

Corollary 9.1 *The symmetric random walk on* $N(G)$ *is* $\epsilon -$ *close to a uniform stationary distribution after* $\frac{2n^3}{\epsilon}$ *steps.*

9.2 What Can Be Done in the General Case

We will define two random walks on $N(G)$ which are 'close' to the symmetric one and converge rapidly. However it is not known to what stationary distribution they converge.

Define a Markov Chain $MC_1(G)$ on $N(G)$ as follows:

1. Given a state $M \in N(G)$, choose a vertex $v \in V$ uniformly at random.
2. If $v \in M$ then with probability $1/2$ move to $M - v$, and with probability $1/2$ stay at M.
3. If $v \notin M$ then with probability $1/2$ stay at M and with probability $1/2$ do -

- If $\Gamma_M(v) = \emptyset$, then move to $M + v$.
- If $\Gamma_M(v) \neq \emptyset$, then let m be the smallest in $\Gamma_M(v)$ (by the order defined on V). Move to $M - m$ (notice that this is the only change from $MC(G)$).

Lemma 9.3 *The probabilities defined by the Markov chain $MC_1(G)$ are -*

- $p_{AB} = \frac{1}{2n}$ *if $A + x = B$ for some $x \in V$.*
- $\frac{1}{2n} \leq p_{AB} \leq \frac{1+d_x}{2n}$ *if $A - x = B$ for some $x \in V$, where d_x is the degree of x.*
- $p_{AB} = 0$ *else.*

To prove rapid convergence of $MC_1(G)$, let us define the following Coupling:

1. Let X_t^1 and X_t^2 be two dependent processes, each behaving as defined by the following walk. Assume they are in states A and B respectively, and that $|A \oplus B| = k$. Both processes choose uniformly at random the same $v \in V$.
2. If $v \in A$ and $v \in B$ then with probability $1/2$, X_t^1 moves to $A - v$ and X_t^2 moves to $B - v$. Clearly $|A \oplus B| = k$.
3. If $v \notin A$ and $v \notin B$ then -
   - If $\Gamma_A(v) = \Gamma_B(v) = \emptyset$, then with probability $1/2$, X_t^1 moves to $A + v$ and X_t^2 moves to $B + v$. Again $|A \oplus B| = k$.
   - If $\Gamma_A(v) = \emptyset$ and $\Gamma_B(v) \neq \emptyset$, then let b be the smallest vertex in $\Gamma_B(v)$. With probability $1/2$ X_t^1 moves to $A + v$ and X_t^2 moves to $B - b$. We have $|A \oplus B| = k + 1 - 1 = k$.
   - If $\Gamma_A(v), \Gamma_B(v) \neq \emptyset$, then let a be the smallest vertex in $\Gamma_A(v)$ and let b be the smallest vertex in $\Gamma_B(v)$. With probability $1/2$ X_t^1 moves to $A - a$ and X_t^2 moves to $B - b$. The distance between A and B can only reduce since -
     - If $a = b$ then $|A \oplus B| = k$.
     - If $a \neq b$, $a \notin \Gamma_B(v)$ and $b \notin \Gamma_A(v)$ then $|A \oplus B| = k - 2$.
     - If $a \neq b$, $a \in \Gamma_B(v)$ and $b \notin \Gamma_A(v)$ then $|A \oplus B| = k - 1 + 1 = k$.
     - The case $a \neq b$, $a \in \Gamma_B(v)$ and $b \in \Gamma_A(v)$ is not possible, since a is the smallest in $\Gamma_A(v)$, therefore $a < b$, and b is the smallest in $\Gamma_B(v)$ therefore $b < a$.
4. If $v \notin A$ but $v \in B$, then with probability $1/2$ X_t^2 moves to $B - v$ and X_t^1 does nothing, and with probability $1/2$ X_t^2 does nothing and X_t^1 moves to -
   - $A + v$, if $\Gamma_A(v) = \emptyset$.

- $A - a$, if $\Gamma_A(v) \neq \emptyset$, where a is the smallest vertex in $\Gamma_A(v)$.

In either case $|A \oplus B| = k - 1$.

It is easy to verify that the above description is indeed a Coupling of $MC_1(G)$. In order to bound the expected Coupling time, let us first prove the following lemma:

Lemma 9.4 *For any two sets $A, B \in N(G)$, it is possible to choose $v \in V$ in step (1) of the coupling defined above, so as to reduce the distance $|A \oplus B|$.*

Proof: Choose $v \in A \oplus B$. Without loss of generality assume $v \in A$ and $v \notin B$ (If no such v exists then $A = B$). By step (4) of the coupling above, $|A \oplus B|$ is reduced by one. □

Theorem 9.3 *The expected coupling time of $MC_1(G)$ is $\leq 2n^2$.*

Proof: Follows from the above lemma and theorem 6.5. □

Corollary 9.2 *The random walk $MC_1(G)$ will be $\epsilon -$ close to its stationary distribution π after $\frac{2n^2}{\epsilon}$ simulation steps.*

We proved that $MC_1(G)$ converges fast, but the question is to what stationary distribution? That is, is the stationary distribution π close to uniform? We will try to bound π (see section 5):

Lemma 9.5 *Let $\pi = (\pi_1, ..., \pi_{N(G)})$ be the stationary distribution of $MC_1(G)$. Then $\frac{1}{n} \leq \frac{\pi_A}{\pi_B} \leq n$ for any two sets $A, B \in N(G)$ such that $|A \oplus B| = 1$.*

Proof: Define $p^*_{AB} = Pr(X_t = B | X_{t+1} = A)$. Then

$$p^*_{AB} = \frac{\pi_B p_{BA}}{\pi_A}$$

and

$$p^*_{BA} = \frac{\pi_A p_{AB}}{\pi_B}$$

Therefore

$$p^*_{AB} p^*_{BA} = p_{AB} p_{BA}$$

But $|A \oplus B| = 1$ and therefore with out loss of generality $A = B + v$ for some vertex $v \in V$. Hence $p_{BA} = \frac{1}{n}$ and $\frac{1}{n} \leq p_{AB} \leq \frac{c}{n}$. Thus

$$\frac{1}{n^2} \leq p^*_{AB} p^*_{BA} \leq \frac{c}{n}$$

Therefore $\frac{1}{n^2} \leq p^*_{AB} \leq 1$. From the definition of p^*_{AB} we get that $\frac{\pi_B}{\pi_A} = \frac{p^*_{AB}}{p_{BA}}$ and hence

$$\frac{1}{n} \leq \frac{\pi_A}{\pi_B} \leq n$$

□

Corollary 9.3 *Let d be the diameter of the underlying graph of $MC_1(G)$. Then $\frac{\pi_A}{\pi_B} \leq n^d$, for any two independent sets $A, B \in N(G)$.*

A slightly different random walk on $N(G)$ can be defined by a Markov Chain $MC_2(G)$:

1. Given a set $M \in N(G)$, choose a vertex $v \in V$ uniformly at random. Let c be the maximum degree of G, and d_v the degree of v.
2. If $v \in M$ has degree $< c$, then with probability $\frac{1}{2} \cdot \frac{c-d_v}{c}$ delete v .
3. If $v \notin M$ then
   - If $|\Gamma_M(v)| = 0$, then with probability $1/2$ add v.
   - If $|\Gamma_M(v)| \neq 0$, then with probability $\frac{1}{2}$ delete a vertex from $\Gamma_M(v)$, each one with equal probability.

Lemma 9.6 *The probabilities defined by $MC_2(G)$ are-*

- $p_{AB} = \frac{1}{2n}$ *if $A + x = B$ for some $x \in V$.*
- $\frac{1}{2n} \leq p_{AB} \leq \frac{c}{2n}$ *if $A - x = B$ for some $x \in V$*

Thus for a graph with maximum degree $c = 3$, the transition probabilities are bounded by $\frac{1}{2n} \leq p_{AB} \leq \frac{3}{2n}$, which is very close to uniform. $MC_2(G)$ can be shown to converge rapidly using the Coupling technique. The proof is omitted.

Further Questions :

1. Can we improve the bound in lemma 9.5 to $\frac{1}{c} \leq \frac{\pi_A}{\pi_B} \leq c$, where c is the maximum degree of G. If so then for any $A, B \in N(G)$ we will get a bound of $\frac{1}{c^d} \leq \frac{\pi_A}{\pi_B} \leq c^d$, where c is as above and d is the maximum distance between any two independent sets. Thus for graphs with independent sets of maximum size $\log n$ we will get a bound of polynomial size in n, and therefore as a result the existence of an f.p. approximate counter of independent sets of such graphs.

2. For what graphs is it true that the stationary distribution of $MC_1(G)$ and $MC_2(G)$ is bounded by $\frac{1}{poly(n)N(G)} \leq \pi_i \leq \frac{poly(n)}{N(G)}$.

3. For what graphs will the symmetric random walk on the Set of independent sets converge rapidly ?

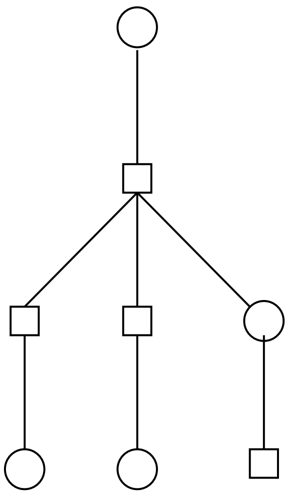
4. Is there a f.p. randomized approximate counter for the number of all independent sets of a graph G ????

10 A deterministic Algorithm for Counting Independent Sets in Trees

For graphs which are trees, it is possible to compute deterministicly, the number of independent sets and maximum independent sets in polynomial time in the number of vertices of the graph. The following lemma shows that this result is not trivial, in a sense that there are trees with many maximum independent sets. Hence constructing them sequentially will take a long time, and therefore a different approach is needed.

Lemma 10.1 *There exists a tree with exponentially many maximum independent sets.*

Proof: Consider for example the following tree, where the number of nodes is n:



It has $2^{\frac{n-2}{2}} + 1$ maximum independent sets, one of them displayed above. $\square$

Lemma 10.2 *Let G be a tree. It is easy to find a maximum independent set of G.*

Proof: Construct a maximum independent set M in the following way: Start at the leaves and add all of them to M. Next add (if possible) each node which has shortest distance 3 from the leaves (the leaves will be said to have distance 1 from the leaves). Continue for nodes with shortest distance $2k+1$, $\quad k = 2, 3, \ldots$ as long as possible. It is clear M is independent. For each node x with distance k from the leaves, let $T(x)$ be the subtree having x as a root. Then it is possible to show by induction on k that M is maximum in $T(x)$ and thus in G. $\square$

Let

$$O(G) = \{x|x \textit{ can not be in any maximum independent set of } G\}$$
$$I(G) = \{x|x \textit{ is in all maximum independent sets of } G\}$$

Lemma 10.3 *Let G be a tree. It is possible to find $O(G)$ and $I(G)$ in time $O(n)$.*

Proof: Let M be the maximum independent set constructed in the lemma 9.2. We will construct two sets O and I, and then show that $I = I(G)$ and $O = O(G)$. For each node x with degree ≥ 3 determine:

- If $x \in M$ then $x \in I$.
- If $x \notin M$ has at least two neighbors in M, then $x \in O$.

For each path between two nodes x, y of degree ≥ 3 do :

- If the path is of odd length, and if $x, y \in I$ or $x, y \in O$, then for each node z along the path, if $z \notin M$ determine $z \in O$ and if $z \in M$ determine $z \in I$.
- If the path is of even length, and if $x \in I$ and $y \in O$, then for each node z along the path, if $z \notin M$ determine $z \in O$ and if $z \in M$ determine $z \in I$.

For each path starting at a leaf and ending before a node x of degree ≥ 3 do:

- If the path is of odd length, and if $x \in O$, then for each node z along the path, if $z \notin M$ determine $z \in O$ and if $z \in M$ determine $z \in I$.
- If the path is of even length, and if $x \in I$ then for each node z along the path, if $z \notin M$ determine $z \in O$ and if $z \in M$ determine $z \in I$.

Claim 10.1 *$O(G) \subseteq O$.*

Proof: Let x be a node of degree ≥ 3, then if $x \in O(G)$ then $x \notin M$. If none of the neighbors of x are in M then M is not maximum. If only one neighbor y of x is in M then we can move to the maximum independent set $M - y + x$ and therefore $x \notin O(G)$. Therefore there must exist at least two neighbors y, z of x in M, and thus $x \in O$. For nodes x of degree ≤ 2 it can be shown in a similar way that if $x \in O(G)$ then $x \in O$.

Claim 10.2 *$I(G) \subseteq I$*

Proof: If $x \in I(G)$ then $x \in M$ and therefore $x \in I$.

Claim 10.3 *$I \subseteq I(G)$.*

Proof: Let $x \in I$, and assume $x \notin I(G)$. But $x \in I$, and therefore the distance of x from the leaves is $2k+1$. Hence, if we delete x, we will be left with a path of length $2k$, from which we can include at most k nodes in any independent set. If we take these k nodes starting from the leaves, we could add x and still remain with an independent set. Contradiction. Thus $x \in I(G)$.

Claim 10.4 *$O \subseteq O(G)$.*

Proof: Similar to the proof of claim 9.3.

Thus $O(G) = O$ and $I(G) = I$, and since constructing I and O required constructing M in time n and then at most two scannings of the tree, the lemma is proved. □

Lemma 10.4 *Let G be a tree on n nodes. There exists a node x, such that each subtree of x has $\leq \frac{n}{2}$ nodes.*

Proof: Let x be a random node of G, and let x_i be its neighbors. Then there exists at most one x_i such that the subtree that has x_i as a root has more then $\frac{n}{2}$ nodes. But then all of the other subtrees have together at most $\frac{n}{2}$ nodes. Move to x_i and continue recursively. □

Lemma 10.5 *Let G be a tree on n nodes. It is possible to find the number of all independent sets of G in time $O(n^2)$.*

Proof: Let $N(G)$ be the number of all independent sets of G, and let x be a node as in the above lemma. Let T_i be its subtrees, and let x_i be the roots of T_i respectively. Then

$$N(G) = \Pi_i N(T_i) + \Pi_i N(T_i - x_i)$$

Continue recursively with T_i. The complexity of the algorithm is:

$$T(n) \leq 4T(\frac{n}{2}) = O(n^2)$$

□

Lemma 10.6 *Let G be a tree on n nodes. It is possible to find the number of maximum independent sets of G in time $O(n^2)$.*

Proof: Let $N(G)$ be the number of maximum independent sets in G. Find $I(G)$ and $O(G)$, and let x be the node as in lemma 10.4. Let T_i be its subtrees, and let x_i be the neighbors of x in each of the subtrees respectively. Then:

- If $x \in O(G)$ then $N(G) = \Pi N(T_i)$.
- If $x \in I(G)$ then $N(G) = \Pi N(T_i - x_i)$.
- If $x \notin O(G), I(G)$ then $N(G) = \Pi N(T_i) + \Pi N(T_i - x_i)$.

Continue recursively with T_i. The complexity of the algorithm is

$$T(n) \leq 4T(\frac{n}{2}) + n = O(n^2)$$

$\square$

11 Partitions

Let n be a positive integer. A *partition* of n is a representation of n as a sum of positive integers:

$$n = x_1 + x_2 + ... + x_k \quad , \quad x_i > 0$$

Let P_{kn} be the set of unordered partitions of n into k parts, that is the set of solutions of

$$n = x_1 + x_2 + ... + x_k \quad x_1 \geq x_2 \geq ... \geq x_k \geq 1$$

Let P_n be the set of unordered partitions of n into any number of parts, that is the set of solutions of

$$n = x_1 + x_2 + ... + x_n \quad x_1 \geq x_2 \geq ... \geq x_n \geq 0$$

An efficient algorithm for computing $|P_{kn}|$ and $|P_n|$ is not known. The asymptotic value of $|P_n|$ was shown to be (see [Ha]):

$$\frac{e^{\pi\sqrt{n\frac{2}{3}}}}{4n\sqrt{3}}$$

Let us now try to estimate $|P_{kn}|$ and $|P_n|$, using the Markov chain method. If we could construct a Markov chain which converges rapidly to a distribution which is close to uniform, then we will be able to get an (ϵ, δ) approximation to $|P_{kn}|$ and $|P_n|$, for any choice of ϵ and δ. We will define two Markov chains on the sets P_{kn} and P_n, and show that they converge rapidly to their stationary distribution, using the Coupling technique. Unfortunately the stationary distribution of these two chains is far from uniform, and therefore the chains can not serve as polynomial generators. We will try to overcome this problem by defining a different Markov chain with distribution which is close to uniform. Its rate of convergence is still not known, but if it was shown to converge rapidly, a direct consequence will be the existence of a f.p. (ϵ, δ)-approximation algorithm for the number of partitions of a given number. These results are presented in the next two subsections.

11.1 A Random Walk on P_n

Let $x = x_1, ..., x_n$ and $y = y_1, ..., y_n$ be two partitions of n. Then the distance between them is defined to be:

$$d(x, y) = \min_{\sigma_1, \sigma_2} \frac{1}{2} \sum_{i=1}^{n} |x_{\sigma_1(i)} - y_{\sigma_2(i)}|$$

where the minimum is taken over all permutations σ_1, σ_2. The minimum is achieved if we take the permutations σ_1 and σ_2, such that $x_{\sigma_1(i)} = y_{\sigma_2(i)}$ for a maximum number of $i's$, and the unmatched parts are ordered by increasing order. It is easy to verify that $d(x, y) \leq n - 1$

for any two partitions x, y of n. The upper bound is achieved if we take $x = 0, 0, ..., 0, n$ and $y = 1, 1, ..., 1$.

We would like to study the following Markov Chain, $MC(n)$, defined on P_n:

Given a partition $x \in P_n$, choose independently at random two indices $1 \le i, j \le n$ and with probability $1/2$, move (if $x_i > 0$) to the partition x^{ij}, such that-

- $x_i^{ij} = x_i - 1$.
- $x_j^{ij} = x_j - 1$.
- $x_k^{ij} = x_k$ for $k \ne i, j$.

Let n_i^x denote the number of parts $x_j = i$ in a partition x.

Then the probability that we move from partition x to $y = x^{ij}$ is:

$$p_{xy} = \frac{n_i^x n_j^x}{2n^2}$$

Theorem 11.1 *The Markov chain $MC(n)$ is reversible, with stationary distribution $\pi = (\pi_1, ..., \pi_{|P_n|})$, such that*

$$\pi_x = \frac{\binom{n}{n_0^x, ..., n_n^x}}{\binom{2n-1}{n}}$$

Proof: We will show that

$$\pi_x = \frac{\binom{n}{n_0^x, ..., n_n^x}}{\binom{2n-1}{n}}$$

is a solution to the set of equations:

$$\pi_x p_{xy} = \pi_y p_{yx}$$

and that

$$\sum_{x \in P_n} \pi_x = 1.$$

Then since the stationary distribution is unique, it will follow that π is indeed the stationary distribution of $MC(n)$, and furthermore that $MC(n)$ is reversible.

Let x, y be two partitions of n, such that $y = x^{ij}$. Then

$$p_{xy} = \frac{n_i^x n_j^x}{2n^2}$$

Hence

$$p_{yx} = \frac{n_{i-1}^y n_{j+1}^y}{2n^2} = \frac{(n_{i-1}^x + 1)(n_{j+1}^x + 1)}{2n^2}$$

And if

$$\pi_x = \frac{\binom{n}{n_0^x, ..., n_n^x}}{\binom{2n-1}{n}}$$

then

$$\pi_y = \frac{\binom{n}{n_0^y, ..., n_n^y}}{\binom{2n-1}{n}} = \frac{\binom{n}{n_0^x, ..., n_{i-1}^x + 1, n_i^x - 1, ..., n_j^x - 1, n_{j+1}^x + 1, ..., n_n^x}}{\binom{2n-1}{n}}$$

Given the above, it is easyly verified that

$$\pi_x p_{xy} = \pi_y p_{yx}.$$

We will now show that

$$\sum_{x \in P_n} \binom{n}{n_0^x, ..., n_n^x} = \binom{2n-1}{n}$$

and therefore

$$\sum_{x \in P_n} \pi_x = 1$$

But this is clear, since each partition $x \in P_n$ defines $\binom{n}{n_0^x, ..., n_n^x}$ ordered partitions of n into n parts, each one with n_i^x parts of size i, $0 \leq i \leq n$. Thus $\sum_{x \in P_n} \binom{n}{n_0^x, ..., n_n^x}$ is the number of all ordered partitions of n into n parts, and this is known to be $\binom{2n-1}{n}$. □

To prove rapid convergence of $MC(n)$, let us define the following **Coupling**:

1. Assume we are in states x and y, and without loss of generality assume they are ordered such that $d(x, y)$ is minimal. Choose uniformly at random two indices i, j.

2. If $x_i = y_i$ or $x_j = y_j$, then with probability 1/2, move to x^{ij} and y^{ij} (if possible).

3. Else with probability 1/2, move to x^{ij} and remain at y, and with probability 1/2, move to y^{ij} and stay at x.

It is easy to verify that this is indeed a coupling of $MC(n)$, because each process is distributed as $MC(n)$, and once they meet at some state they continue together. We want to show now that the coupling time is polynomial:

Lemma 11.1 *In each step of the coupling above there is probability $p \geq \frac{1}{2n^2}$ to reduce the distance between the two processes, and probability $q \leq p$ to increase it.*

Proof: Let $x, y \in P_n$, and assume that $d(x, y) = d$. Then since $x \neq y$, there must exist indices i, j, such that $x_i > y_i$ and $x_j < y_j$. If we move to x^{ij} and remain at y, the new distance will be $\leq d - 1$, and therefore $p \geq \frac{1}{2n^2}$, and if we move to y^{ij} and remain at x, the new distance will be $\leq d + 1$. For a choice of indices i, j, such that either $x_i = y_i$ or $x_j = y_j$, the distance can only reduce. Therefore $p \geq q$. □

Hence:

Theorem 11.2 *The expected coupling time of $MC(n)$ is $\leq n^4$.*

Proof: For any two partitions $x, y \in P_n$, we have that $d(X, Y) \leq n - 1$. Therefore by theorem 6.5 and the above lemma, the expected coupling time is $\leq \frac{(n-1)^2}{2p} \leq n^2(n-1)^2 \leq n^4$. □

Corollary 11.1 *The random walk $MC(n)$ will be $\epsilon -$ close to its stationary distribution π, after $\frac{n^4}{\epsilon}$ simulation steps.*

11.2 Random Walks on P_{kn}

A similar Markov chain $MC(n, k)$, can be defined on P_{kn}:

Given a partition $x \in P_{nk}$, choose independently at random two indices $1 \leq i, j \leq k$ and with probability 1/2, move (if $x_i > 1$), to the partition x^{ij}, such that-

- $x_i^{ij} = x_i - 1$.
- $x_j^{ij} = x_j + 1$.
- $x_k^{ij} = x_k$ for $k \neq i, j$.

Then using the notation of the above section we have-

Theorem 11.3 *The Markov chain $MC(n,k)$ is reversible, with stationary distribution $\pi = (\pi_1, ..., \pi_{|P_n|})$, such that*

$$\pi_x = \frac{\begin{pmatrix} k \\ n_1^x, ..., n_{n-k-1}^x \end{pmatrix}}{\begin{pmatrix} n-1 \\ k-1 \end{pmatrix}}$$

Proof: Similar to the proof of theorem 11.1. □

Theorem 11.4 *The expected coupling time of $MC(n,k)$ is $\leq (n-k-1)^2 k^2$.*

Proof: For any two partitions $x, y \in P_{nk}$, the distance is $d(x,y) \leq n-k-1$. Define a similar coupling as the one of $MC(n)$. Then there is a probability of $p \geq \frac{1}{2k^2}$ to reduce $d(x,y)$, and probability $q \leq p$ to increase it. Therefore by theorem 6.5, the expected coupling time is $\leq (n-k-1)^2 k^2$. □

Corollary 11.2 *The random walk $MC(n,k)$ will be $\epsilon -$ close to its stationary distribution π, after $\frac{(n-k-1)^2 k^2}{\epsilon}$ simulation steps.*

Thus, although the two random walks $MC(n)$ and $MC(n,k)$ converge rapidly to their stationary distribution, they can not be used as generators in polynomial counting procedures for P_n and P_{kn} respectively, because their stationary distributions are not close to uniform. As a possible solution, let us define a different walk on P_{kn} with a stationary distribution which is close to uniform.

Describe a partition of n as a vector P of length n, such that $P[i] = j$ if there are j $i's$ in the sum representing n. Thus $P \in P_{kn}$ if $\sum P[i] = k$ and $\sum iP[i] = n$, and $P \in P_n$ if $\sum iP[i] = n$.

Define a distance on the set of vectors by $d(P_1, P_2) = \sum_{i=1}^{n} |P_1[i] - P_2[i]|$.

Two vectors $P, P' \in P_{kn}$ are *neighbors* if there exist indices i, j, m, l, such that:

1. $i + j = m + l$.
2. $P[i] - 1 = P'[i],\ P[j] - 1 = P'[j],\ P[m] + 1 = P'[m],\ P[l] + 1 = P'[l]$.
3. $P[t] = P'[t]$ for any $t \neq i, j, m, l$.

Define a random walk on P_{kn}:

Given a vector $P \in P_{kn}$, choose uniformly at random one of its neighbors and with probability 1/2 move to it.

Lemma 11.2 *The Markov Chain defined by this walk is ergodic.*

Proof: The chain is clearly aperiodic, since there is probability $1/2$ to remain at any given state. We will show that it is also irreducible and hence ergodic. Let $P_1, P_2 \in P_{kn}$, such that $P_1 \neq P_2$. Then there exist indices i, m, such that $P_1[i] > P_2[i]$ and $P_1[m] < P_2[m]$. There also must exist a third index j, such that $P1[j] \neq P_2[j]$. Else

$$iP_1[i] + mP_1[m] = iP_2[i] + mP_2[m] \quad , \quad P_1[i] + P_1[m] = P_2[i] + P_2[m]$$

But from this two equations we get that -

$$(P_1[m] - P_2[m])(m - i) = 0$$

and since $P_1[m] - P_2[m] \neq 0$ we get that $i = m$. Contradiction. Therefore there exists such a j, and without lost of generality assume $P_1[j] > P_2[j]$. It can also be shown that $i < m < j$ or $j < m < i$, and so there exists an index l such that $i + j = m + l$. Then if P_1 will move to the neighbor defined by i, j and m, l, the distance between P_1 and P_2 will be reduced by at least 2. Hence there is a path of length $\leq \frac{d(P_1,P_2)}{2}$ between any two vectors P_1, P_2, and so the chain is irreducible. $\square$

Lemma 11.3 *The Markov chain defined by this walk is reversible, and its stationary distribution $\pi = (\pi_1, ..., \pi_{P_{kn}})$ is bounded by $\frac{1}{2E} \leq \pi_p \leq \frac{n(k-1)}{4E}$ for any $P \in P_{kn}$, where E is the number of edges in the underlying graph of the chain.*

Proof: Each two pairs i, j and m, l such that $i + j = m + l$ define a unique transition from P to P', with probability $\frac{1}{2d_p}$, where d_p is the number of possible transitions from P. Furthermore if i, j and m, l define a transition from P to P', then m, l and i, j define a transition from P' to P. Therefore the chain is reversible, and its stationary distribution is $\pi_p = \frac{d_p}{2E}$. For each $P \in P_{kn}$ there are at most k indices i, such that $P(i) > 0$. Let them be $i_1, ..., i_k$. For each pair i_s, i_t we can choose at most $\frac{i_s+i_t}{2}$ pairs m, l such that $i_s + i_t = m + l$. Therefore

$$d_p \leq \sum_{t=1}^{k-1} \sum_{s=t+1}^{k} \frac{i_s + i_t}{2} = \frac{(k-1)(i_1 + ... + i_k)}{2} = \frac{n(k-1)}{2}$$

and so $\pi_p \leq \frac{n(k-1)}{4E}$. The lower bound on π_i is trivial, since the chain is irreducible and so $d_p \geq 1$. $\square$

The rate of convergence of this last Markov chain is not known so far, but if it was shown to be rapidly converging, we will have as a result an f.p. approximate counter for partitions of n into k parts.

12 Acknowledgements

I am very grateful to my advisor Avi Wigderson for suggesting this subject, for his encouragement and unfailing support and for many helpful comments throughout this work. I would also like to thank Ilan Newman for many fruitful discussions, and a special thanks to Oded Goldreich whose remarks made this work more readable.

13 References

[A1] D.Aldous, Random Walks on Finite Groups and Rapidly Mixing Markov Chains. *Seminaire de Probabilites XVII.*

[A2] D.Aldous, On the Markov Chain Simulation Method for Uniform Combinatorail Distributions and Simulated Annealing. *Technical Report No. 60. Department of Statistics, University of California.*

[B] A.Z.Broder, How Hard is it to Marry at Random? (On the Approximation of the Permanent). *Proc. of the 18 Ann. Symp. on Theory of Computing.*

[D] P.Dagum, On the Magnification of Exchange Graphs with applications to Enumaration Problems. *Phd. Thesis, 1989.*

[G] D.Griffeath, A maximal Coupling for Markov Chains. *Z. Wahrscheinlichkeitstheorie Verw. Gebiete 31, 1975.*

[Ha] M.Hall, Combinatorial Theory. *A Wiley Interscience publication.*

[Ho] M.Hofri, Probabilistic Analysis of Algorithms. *Springer Verlag, New York, 1987.*

[JVV] M.R.Jerrum, L.G.Valiant and V.V.Vazirani, Random Generation of Combinatorial Structures from a Uniform Distribution. *Theoretical Computer Science 43.*

[JS] M.Jerrum and A.Sinclair, Conductance and the Rapid Mixing Property for Markov Chains : the Approximation of the Permanent Resolved. *Proc. of hte 20 Ann. Symp. on Theory of Computing.*

[KL] R.M.Karp and M.Luby, Monte-Carlo Algorithms for the Planar Multiterminal Network Reliability Problem. *Journal of Complexity, volume 1, number 1, 1985.*

[KLM] R.Karp,M.luby and N.Madras, Monte - Carlo Approximation Algorithms for Enumeration problems. *Technical Report 203/87, Department of Computer Science, University of Toronto.*

[KKLLL] N.Karmarkar,R.Karp,R.Lipton,L.Lovasz and M.luby, A Monte - Carlo Algorithm for Estimating the Permanent. *Theoretical Computer Science, 43, 1986.*

[KS] J.Kemeny and J.Snell, Finite Markov Chains.

[MV] M.Mihail and U.Vazirani, Conductance and Convergence of Markov Chains.

[P] J.W.Pitman, On Coupling of Markov Chains. *Z. Wahrscheinlichkeitstheorie Verw. Gebiete 35, 1976.*

[PB] J.S.Provan and M.O.Ball, The Complexity of Counting Cuts and of Computing the Probability that a Graph is Connected. *SIAM. J. of Computing 12.*

[Ro] S.Ross, Stochastic Processes.

[Ry] H.J.Ryser, Combinatorial Mathematics. *The Mathematical Association of America, no. 14, 1963.*

[S] C.P.Schnorr, Optimal Algorithms for Self-Reducible Problems. *Proceedings of the 3 International Colloquium on Automata, Languages and Programming, 1976.*

[SJ] A.Sinclair and M.Jerrum, Approximate Uniform Generation and Rapidly mixing Markov Chains. *Internal Report CSR-241-87, Department of Computer Science, University of Edinburgh.*

[V] L.Valiant, The Complexity of Enumeration and Reliability Problems. *SIAM J. of Computing 8.*