

**Linear and Non-linear Systems:
A survey**

Josep Diaz
Maria Serna
Paul Spirakis

Report LSI-97-56-R

Linear and Non-linear Systems: A survey *

J. Diaz[†] M. Serna[†] P. Spirakis[‡]

November 19, 1997

Abstract

In this paper we present the research that has been done with Linear Dynamical Systems to generate almost uniformly elements from a given set, and thus approximate some hard counting problems. We also indicate how non-linear systems can help to parallelize the computation. We end, presenting further applications of linear systems to formalize heuristics.

1 Introduction

Many problems involving the counting of the number of solutions of combinatorial structures are well known to be difficult. Valiant defined the class $\#P$ of computationally equivalent counting problems ([Val79b]). For many problems in this class, their decision counterpart is in P . It is well known that, unless the polynomial hierarchy collapses, $P \neq \#P$. This fact implies that for any $\#P$ -complete problem, exact counting is apparently intractable ([Pap94]). The most notorious of these problems is to compute the permanent of a dense matrix, that turns out to be equivalent to counting the number of perfect matchings in a dense bipartite graph ([Val79a]). The hardness of these counting problems motivated research on approximate counting. Pioneering work in this line was the paper [KLM89] where they construct a Randomized Fully Approximation Scheme for some difficult counting problems. Later, it was discovered that for the problems which are

*This research was partially supported by the ESPRIT Long Term Research Project no. 20244 - ALCOM-IT and CICYT project TIC97-1475-CE.

[†]Departament de Llenguatges i Sistemes, Universitat Politècnica Catalunya, Pau Gargallo 5, 08028-Barcelona {diaz,mjserna}@goliat.upc.es

[‡]Computer Technology Institute (C.T.I.), P.O. Box 1122, 26110 Patras, Greece, spirakis@cti.gr

self-reducible, approximate counting is equivalent to almost uniform generation ([JVV86]). The almost uniform generation problem consist in picking at random an element of a finite set according to some distribution, with a relative error of at most ϵ with respect to the probability that a given element is chosen, under the uniform distribution.

A Monte-Carlo algorithm for approximating the permanent of dense positive matrices based in computing an unbiased estimator, was given by Karmarkar, Karp, Lipton, Lovasz and Luby ([KKL⁺93]). Their estimator can be easily computed in RNC, thus we get a RNC algorithm to approximate the permanent of the adjacency matrix of a bipartite graph with minimum degree $(1/2 + \alpha)n$ where $\alpha > 0$. Therefore there exists a RNC approximation to the number of perfect matchings for “quite dense” bipartite graphs.

A technique that has proved to be very useful for solving the almost uniform generation problem, is the Markov Chain technique. Given a problem, define a Markov chain where the states are all possible solutions, plus possibly a small fraction of “non-solutions”, and the transitions are certain probabilistic rules that allow us to remain in the same state or to pass to a new state. Under certain properties of the underlying graph, it can be proved that a polynomial (in the input size) random walk on the states gives us an almost randomly generated element from the stationary distribution of the chain. The difficulty of this method is to prove convergence in a polynomial number of steps to the stationary distribution, usually referred to as the “rapid mixing” property. Broder used the Markov chain technique to approximate the value of the permanent of a dense matrix ([Bro86]). The rapid mixing property of his chains was shown by Jerrum and Sinclair ([JS89]). Over the past years, a large body of literature has been devoted to the subject of almost uniform generation through Markov chains and methods of proving rapid mixing. Excellent surveys can be found in [Sin93, Vaz91, Kan94].

A question of interest is the possibility of parallelizing the almost uniform generation and approximate counting problems. Consider the Markov Chain defined by Broder ([Bro89]) for almost uniform generation of perfect matchings in dense bipartite graphs. Teng has proved that the problem of computing the final node m' of a sequential walk given that it starts from a node m is P-complete ([Ten95]). This result does not exclude the possibility of generating in parallel an almost uniform perfect matching. The Teng result excludes the possibility of a NC simulation of a given sequential random walk, that is; given the walk and the initial state, compute in NC the final state.

To obtain a parallel generator instead of using a Markov chain, we define

a “genetic system”. Such a system starts from a set S of objects with a given initial distribution Π_0 , this will be the initial generation at time $t = 0$. From that initial population, new generations are grown by mating two randomly selected parents. Define a **mating** rule to crossover objects: our rule will be defined in such a way that for any two objects sampled according to distribution Π_t at time t , form a new object that will be an element of the next population. Formally, if u and v are the objects sampled from Π_t mate them with probability $p(u, v, w)$ to outcome the new element w . Then the probability distribution of the population at time $t + 1$ follows a non-linear dynamic equation

$$\Pi_{t+1}(x) = \sum_u \Pi_t(u) \sum_v p(u, v, x) \Pi_t(v).$$

We shall show that the system evolves towards a unique stationary distribution. This kinds of non-linear equations were previously studied. In general quadratic dynamic systems are difficult and no too much is known about their behaviour. For instance, it is known that a quadratic dynamic system can solve any problem in PSPACE, using a polynomial amount of time [Pud94, ARV94], therefore unless $P=NP$ there is not polynomial time simulation of a general quadratic dynamic system. There are results for some particular ones, for example in the work of Rabani et al. [RSW92, RRS95].

The next section in this paper presents some general results on Markov chain theory and the Markov chain method to sample almost uniformly. We survey the sequential approach to solve a general problem, the monomer-dimer system. Section 3 presents the non-linear approach to parallelize the almost uniform sampling. We show the convergence of the system and how to implement in parallel the defined mating rule. Finally section 4 surveys some of the work done trying to formalize hillclimbing heuristics, together with some final remarks.

2 Linear Systems

We shall review some of the basic concepts of Markov chains. Recall that a **Markov chain** μ is an stochastic process, defined on a set of states S , in terms of a transition matrix $P = (p_{ij})_{i,j \in S}$, where each p_{ij} denotes the probability of going from i to j . Therefore, $\forall i \in S$, $\sum_{j \in S} p_{ij} = 1$. Moreover, at $t = k$, we define π_k as

$$\pi_k(i) = \sum_{j \in S} p_{ji} \cdot \pi_{k-1}(j) \tag{1}$$

On the other hand, let X_t be a stochastic variable such that, at time t , it denotes the state where μ is. Also let the initial distribution $\pi_0(i)$ (at $t = 0$) defined as follows: $\forall i \in S, \pi_0(i) = Pr\{X_0 = i\}$. If $S = \{1, 2, 3, \dots, m\}$ and $\bar{\pi}_k = (\pi_k(1), \dots, \pi_k(m))$ is the distribution at time $t = k$, then

$$\bar{\pi}_k = \bar{\pi}_{k-1} \cdot P = \bar{\pi}_0 \cdot P^k. \quad (2)$$

So every Markov chain defines a linear system and reciprocally any linear system can be viewed as a Markov chain.

A Markov chain is **irreducible** if $\forall i, j \in S, \exists t$ such that $p_{ij}^t > 0$. A Markov chain is **aperiodic** if $\forall i, j \in S, \gcd\{t | p_{ij}^t > 0\} = 1$. A Markov chain is said to be **ergodic** if $\forall j \in S, \lim_{t \rightarrow \infty} p_{ij}^t = \pi_\infty(j) > 0$. If μ is ergodic then $\bar{\pi}_\infty = (\pi_\infty(1), \dots, \pi_\infty(n))$ is called the **stationary distribution**.

Let $\bar{1} = (1, 1, \dots, 1)^\top$. As P is stochastic, then $P \cdot \bar{1} = \bar{1}$. Let also

$$P^\infty = \lim_{t \rightarrow \infty} [P_{ij}^t] = \begin{pmatrix} \pi_\infty(1) & \dots & \pi_\infty(n) \\ \vdots & \ddots & \vdots \\ \pi_\infty(1) & \dots & \pi_\infty(n) \end{pmatrix}$$

The following results could be found in any basic book of Markov chains (see for ex. [Nor97, Sin93])

Theorem 1. *A Markov chain μ is ergodic if and only if it is irreducible and aperiodic. Moreover, if a Markov chain μ is ergodic then its stationary distribution is the unique distribution that satisfies:*

$$\begin{aligned} \bar{\pi}_\infty \cdot P &= \bar{\pi}_\infty \\ \sum_{i \in S} \pi_\infty(i) &= 1. \end{aligned}$$

To assure that the stationary distribution tends to be uniform, we need further conditions on μ . A Markov chain is **symmetric** if $\forall i, j \in S, p_{ij} = p_{ji}$. An ergodic Markov chain is **reversible** if $\forall i, j \in S$ satisfy the Balance Equation

$$\pi_\infty(i) p_{ij} = \pi_\infty(j) p_{ji}.$$

The next two results can be found in any canonical textbook on Markov chains,

Proposition 1. *Let μ be ergodic. If $\exists \bar{\pi}_*$ such that $\forall i, j \in S : \pi_*(i) p_{ij} = \pi_*(j) p_{ji}$ and $\sum \pi_*(i) = 1$, then μ is reversible and $\bar{\pi}_\infty = \bar{\pi}_*$.*

Theorem 2. *If μ is an ergodic Markov chain with $|S| = n$ and it is symmetric then*

$$\forall i \in S, \pi_\infty(i) = \frac{1}{n}.$$

Let us consider the problem of given a large and finite set S , and a probability distribution π on S , sample an element in S according to π . The **Markov Chain Technique** gives an approximate solution of the previous problem, and consists in the following steps: Construct a Markov chain μ with states S and stationary distribution π . Starting from an arbitrary state $s \in S$, perform a random walk in the chain large enough to set a *closed* point to equilibrium distribution. In the light of our previous comments μ must be ergodic and if μ is ergodic and symmetric, then

$$\bar{\pi}_\infty = \left(\frac{1}{|S|}, \dots, \frac{1}{|S|} \right)$$

i.e. $\bar{\pi}_\infty$ is uniform.

Therefore, once we have a Markov chain μ , to sample from π_∞ , simulate the Markov chain for a finite number of steps and get *close* to π_∞ . The question is, what is the *rate of convergence*?. Recall that for all k , $\bar{\pi}_k = \bar{\pi}_0 P^k$, note that we need to control powers of P . Hence, we need to look at the eigenvalues of P . The basic idea is to use spectral theory as it is done in Graph Theory (see for example [Chu96]).

Recall from linear algebra that any $n \times n$ matrix M over K could be considered as a linear operator $V \rightarrow V$. Moreover, if M has $\lambda_1, \lambda_2, \dots, \lambda_n$ real eigenvalues, not necessarily all different, then $M = \sum_{i=1}^n \lambda_i H^i$ with

$$H^i \cdot H^j = \begin{cases} \neq 0 & \text{if } i = j \\ 0 & \text{if } i \neq j. \end{cases}$$

Moreover, if m diagonalizes, then $H^i \cdot H^i = H^i$.

Let P be the transition matrix of μ with eigenvalues $\lambda_1, \dots, \lambda_n$, and assume that P diagonalizes. Then as P is stochastic $\lambda_1 = 1$ and $1 = |\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$. So,

$$P^m = \sum_{i \geq 1} \lambda_i^m H^i = \lambda_1^m H^1 + \sum_{i \geq 2} \lambda_i^m H^i.$$

Let $\bar{\pi} = (\pi(1), \dots, \pi(n))$ be the left eigenvector of $1 = \lambda_1$ and let $|\lambda_2| < 1$ (if $\lambda_2 = -1$ then μ is periodic). Then $\bar{\pi}_\infty = \bar{\pi}$ is the left eigenvector of $\lambda_1 = 1$

and if

$$H^1 = \begin{pmatrix} \pi(1) & \dots & \pi(n) \\ \vdots & \ddots & \vdots \\ \pi(1) & \dots & \pi(n) \end{pmatrix}$$

then

$$\lim_{m \rightarrow \infty} P^m = H^1 + \lim_{m \rightarrow \infty} \sum_{i \geq 2} \lambda_i^m H^i \approx H^1 + \lim_{m \rightarrow \infty} |\lambda_2| H^2.$$

Therefore if μ is an ergodic Markov chain with stationary distribution π_∞ , then $|\lambda_2| < 1$.

To avoid that negative eigenvalues can delay the ratio of convergence, we can increase the value of the self-loop in P and make all eigenvalues positive. The following theorem is proved in [Sin93],

Theorem 3. *If P is the transition matrix of an ergodic and reversible Markov chain with eigenvalues $\lambda_1 = 1 > \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_n > -1$, then the Markov chain with matrix $P' = \frac{1}{2}(I + P)$ is also ergodic and reversible, and it has the same limit distribution π_∞ . The eigenvalues $\{\lambda'_i\}_{i=1}^n$ are similarly ordered and $\lambda'_i = \frac{1}{2}(1 + \lambda_i) > 0$.*

We wish to approach the stationary distribution in a random walk of polynomial length. Define the **relative pointwise distance** at time t is

$$\Delta(t) = \max_{i,j \in S} \frac{|p_{ij}^t - \pi_\infty(j)|}{\pi_\infty(j)}$$

To see how fast $\Delta(t) \rightarrow 0$ and get some bounds on $\Delta(t)$, the following bound is useful for an ergodic and reversible Markov chain, (citeSinclair)

$$|\lambda_2|^t \leq \Delta(t) \leq \frac{|\lambda_2|^t}{\min_{j \in S} \pi_\infty(j)}$$

historically the way to bound the convergence of Markov chains was coupling ([Lin92]). The big breakthrough of Jerrum and Sinclair was to use structural properties of the graphs represented de Markov chains associated to certain counting problems, to bound the convergence. Let us define the concept of rapidly mixing Markov Chain.

The rate of convergence in a Markov chain μ to its stationary distribution π_∞ is given by its mixing time function defined by $\tau_i(\epsilon) = \min\{t \mid \forall t' \geq t : \Delta_i(t') \leq \epsilon\}$.

We say that a Markov chain is **rapidly mixing** if from any $i \in S$ and $\forall \epsilon : 0 < \epsilon \leq 1$ we have

$$\tau_i(\epsilon) \leq \text{poly} \left(|i|, \log \frac{1}{\epsilon} \right).$$

Notice a Markov Chain μ can be considered as a weighted directed graph $G = (V, E, w)$, where V is the set of states, $p_{ij} > 0 \implies (i, j) \in E$, and the weight of an edge $w(i, j)$ is defined as the probability $\pi_\infty(i)p_{ij}$. Notice that when the Markov chain is reversible it holds $w(i, j) = w(j, i)$.

Let us give some topological definitions on the underlying graph G of μ . For S' be a nonempty subset of S . The **capacity** of S' measures the probability of being in a state of S' when reaching π_∞ , and it is defined $C_{S'} = \sum_{i \in S'} \pi_\infty(i)$. The **ergodic flow** of S' measures the probability of leaving S' , and it is defined $F_{S'} = \sum_{i \in S'} \sum_{j \in \bar{S}'} \pi_\infty(i)p_{ij}$. The probability of

leaving S' once inside S' is given by $\Phi_{S'} = F_{S'}/C_{S'}$. The **conductance** of μ is defined as $\Phi = \min\{\Phi_{S'} : S' \subsetneq S, C_{S'} \leq 1\}$. Notice the conductance of μ measures the worst bottle-neck.

If the Markov chain μ is reversible we have

$$\forall S' \subsetneq S : F_S = F_{S'} \implies \Phi = \min \{ \max\{\Phi_{S'}, \Phi_{\bar{S}'}\} : S' \subsetneq S, S' \neq \emptyset \}.$$

Then it is possible to bound λ_2 , and hence the convergence, in terms of the topological characteristic of μ , Φ . The following result is from [JS89],

Theorem 4 (Jerrum-Sinclair). *Let μ be an ergodic and reversible Markov chain. Then*

$$1 - 2\Phi \leq \lambda_2 \leq 1 - \frac{\Phi^2}{2}.$$

Corollary 1. *If μ is ergodic and reversible then*

$$\Delta(t) \leq \frac{(1 - \Phi^2/2)^t}{\min_i \pi_\infty(i)}.$$

Moreover, if $\Phi \leq 1/2$ then $\Delta(t) \geq (1 - 2\Phi)^t$.

The following theorem gives us a characterization of rapid mixing in terms of Φ :

Theorem 5. *Let μ be an ergodic and reversible Markov chain with $\forall i \in S : p_{ii} \leq 1/2$, and such that if $\pi_\infty^* = \min_{i \in S} \{\pi_\infty(i)\}$ then $\ln \pi_\infty^*{}^{-1} \leq \text{poly}(|i|)$. Then μ is rapidly mixing if and only if $\Phi \geq \frac{1}{\text{poly}(|i|)} \forall i \in S$.*

Therefore to prove that an ergodic and reversible Markov chain μ is rapidly mixing, we have to find a polynomial p such that $\Phi \geq 1/p(|i|)$.

Still it is necessary to compute or find bounds for the conductance. For that, Jerrum and Sinclair considered a clever argument to estimate the bottleneck of the Markov chain. Define a unique canonical path, between every pair of states, and given a transition edge count the number of canonical path going through it.

Given any $S' \subseteq S$ let $\delta(S') = \{i \in S' \mid \exists j \in \overline{S'} : e = (i, j)\}$. We define the **edge magnification** of the graph of μ as

$$\gamma(\mu) = \min_{0 < |S'| < \frac{|S|}{2}} \frac{|\delta(S')|}{|S'|}.$$

Notice that if d is the maximum degree of the graph of μ then $\forall S' \subseteq S : |\delta(S')| \geq |S'| \cdot d$. So $0 < \delta(\mu) \leq d$.

Many Markov chains can be considered as a random walk in the graph of μ , with maximum degree d , where transitions from i to j are made with probability β/d for some constant β ($0 < \beta \leq 1$). In addition, $\forall i \in S$, i has a self loop with probability $1 - \beta \deg(i)/d$. In such a situation, the conductance of the corresponding graph verifies $\Phi = \frac{\beta \gamma(\mu)}{d}$. Therefore, in this kinds of Markov chains, to prove they are rapid mixing, it is enough to find a polynomial p such that $\Phi = \frac{\beta \gamma(\mu)}{d} \geq \frac{1}{p(|i|)}$. If $d \leq \text{poly}(|i|)$, we just have to find a polynomial lower bound on the edge magnification γ .

2.1 Monomer-Dimer Systems

Let us see an generic example taking from [Sin93].

Given a graph $G = (V, E)$ with $|V| = n$ and $|E| = \mu$. For $k \in \{0, \dots, \lfloor n/2 \rfloor\}$, let $M_k(G)$ denote the set of matchings of size k in G , and M denote the set of all its matchings, that is $M = \cup_k M_k$. We will also use N to denote the number of matchings, i.e. $N = |M|$. From now on, G will denote the input graph.

We start by defining a Markov chain \mathcal{D} for a given weighted graph G , here $c(e)$ denotes the weight of edge e . The chain \mathcal{D} contains as states the set M of all matchings, and the transitions are defined as follows,

Definition of transitions in \mathcal{D} :

Given a matching $m \in M$,

- (0) Sample uniformly a random edge $e = (u, v)$.

- (1) With probability $1/2$ stay in m
otherwise
 - (1.1) (Deletion) If e is in m then with probability $1/(1 + c(e))$ go to matching $m - \{e\}$, otherwise stay in m .
 - (1.2) (Augmentation) If $m \cup \{e\}$ is a matching then with probability $c(e)/(1 + c(e))$ go to new matching $m \cup \{e\}$.
 - (1.3) (Rotation) If u is unmatched in m and v is matched in m by edge $e' = (v, w)$, then with probability $c(e)/(c(e') + c(e))$ the new matching is $m - \{e'\} \cup \{e\}$, and with probability $c(e')/(c(e') + c(e))$ keep m .
 - (1.4) Otherwise stay in m .

In figure 2.1 it is given an example of the Markov chain corresponding to a monomer-dimer system in which all edge weights are equal to a constant c .

The convergence of the monomer-dimer system can be found in [Sin93]

Theorem 6 (Sinclair). *Given a weighted graph G , the Markov chain D is reversible and ergodic. Moreover the stationary probability $\Pi_\infty(i) = \frac{\prod_{e \in m_i} c(e)}{\sum_{j \in M} \prod_{e \in m_j} c(e)}$.*

In order to analyze the mixing time of such chain, using the conductance argument, we suppose that there is an underlying order on all simple paths (including cycles) in G . Fix for any path a start vertex, that must be an endpoint if it is not a cycle. For any two given matchings m_1 and m_2 their symmetric difference is a set of disjoint cycles and paths. We start by sorting this set of paths according to the fixed order. The canonical path from m_1 to m_2 involves the transformation of the initial matching into the final one, by modifying in order all the paths and cycles, starting from the corresponding start vertices.

To unwind a path that is not a cycle, we have two cases, first, the path starts with an edge of m_2 , in such a case we change edge by edge and finish adding the last if necessary. Second, the path starts with an edge of m_1 remove that edge and follow as in the previous case. An example of such unwinding is given in figure 2

To unwind a cycle fix a direction to traverse its edges in such a way that the first edge from the start vertex is in m_1 remove this edge and proceed as in the corresponding path using as start vertex the endpoint that is not the start vertex of the actual path.

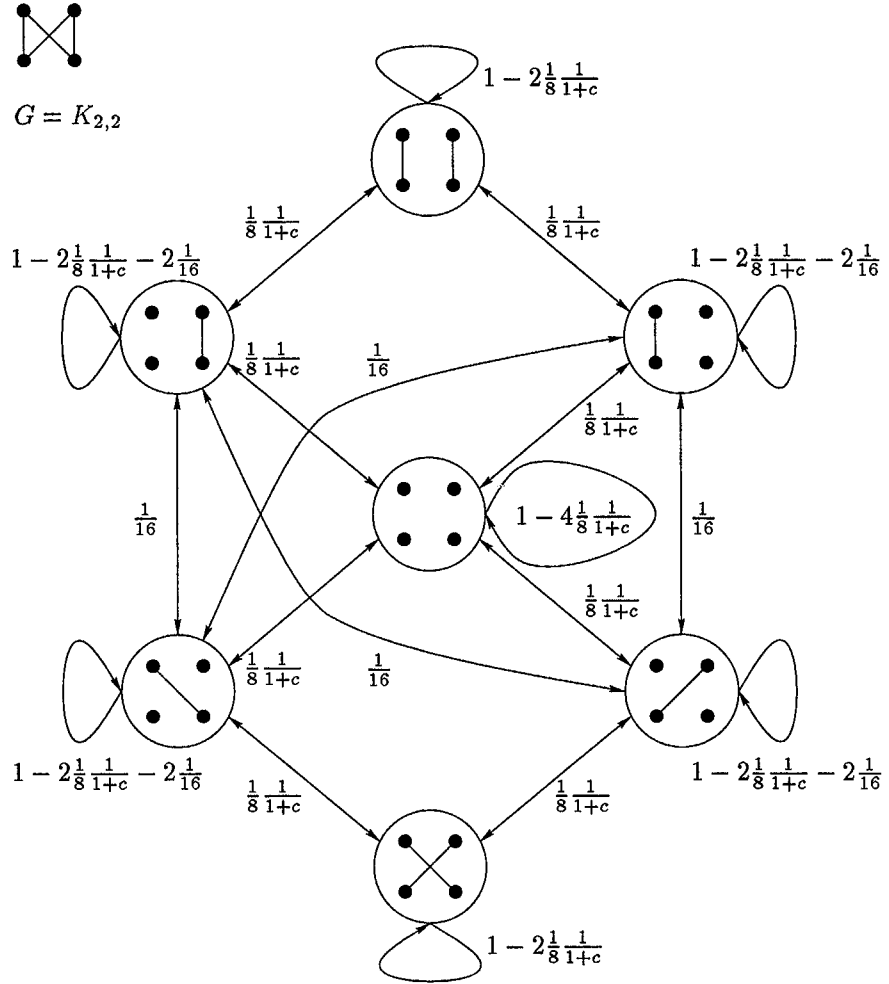


Figure 1: Example with $G = K_{2,2}$ and $|E| = 4$

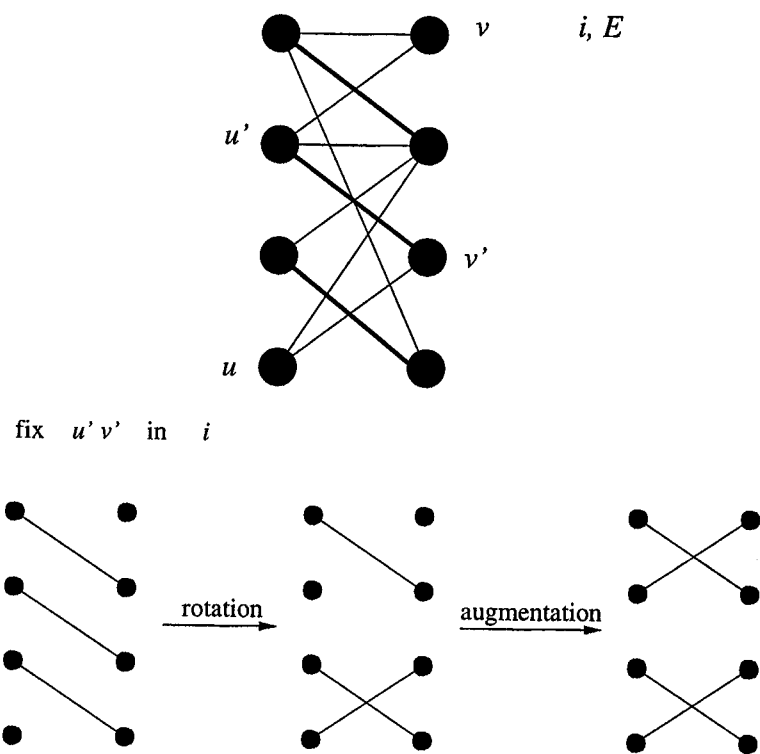


Figure 2: An example of canonical path.

The second part is to define an injective mapping from the set of canonical paths that pass through a given transition t . Let t be a transition from matching m to matching m' , and let $P(t)$ be the set of canonical paths that contain t . For any pair of matchings such that the canonical path from m_1 to m_2 goes through t define $\sigma_t(m_1, m_2) = m_1 \oplus m_2 \oplus (m \cup m')$, and remove the edge e of m_1 adjacent to the start vertex of the path currently treated in case the resulting set of edges is not a matching. The difference $m_1 \oplus m_2$ can be recovered from $\sigma_t(m_1, m_2)$ using

$$m_1 \oplus m_2 = \begin{cases} \sigma_t(m_1, m_2) \oplus (m \cup m') \oplus e & \text{if } t \text{ is an augmentation} \\ & \text{the current path is a cycle} \\ & \text{and } e \text{ is the removed edge} \\ \sigma_t(m_1, m_2) \oplus (m \cup m') & \text{otherwise} \end{cases}$$

It is possible to tell whether the path is a cycle or not, because we are unwinding cycles in different direction than paths. We can recover the original matchings using the path ordering. Hence σ_t is injective. Furthermore it can be shown (see [Sin93])

Lemma 1. *For any transition t and any $\langle m_1, m_2 \rangle \in P(t)$ we have*

$$\pi_\infty(m_1)\pi_\infty(m_2) \leq 4|E|c_{\max}^2 w_t \pi_\infty(\sigma_t(m_1, m_2)),$$

where $c_{\max} = \max\{1, \max_{e \in E} c(e)\}$ and w_t and w_t is $\pi_\infty(m)$ multiplied by probability of the transition t .

Therefore the chain is rapidly mixing.

3 Genetic System

To simplify the presentation, we only consider the monomer-dimer Markov chain \mathcal{M} in the case that all weights are equal to a given fixed parameter $c > 0$. Such chain \mathcal{M} contains as states the set M of all matchings, and the transitions are defined as follows,

Definition of transitions in \mathcal{M} :

Given a matching $m \in M$,

- (0) Sample uniformly a random edge $e = (u, v)$.
- (1) With probability $1/2$ stay in m
otherwise

- (1.1) If e is in m then with probability $1/(1+c)$ go to matching $m - \{e\}$, otherwise stay in m .
- (1.2) If $m \cup \{e\}$ is a matching then with probability $c/(1+c)$ go to new matching $m \cup \{e\}$.
- (1.3) If u is unmatched in m and v is matched in m by edge $e' = (v, w)$, then with probability $1/2$ the new matching is $m - \{e'\} \cup \{e\}$, and with probability $1/2$ keep m .
- (1.4) Otherwise stay in m .

As \mathcal{M} is a restricted version of the monomer-dimer system we have

Theorem 7 (Sinclair). *Given a fixed c , the Markov chain \mathcal{M} is reversible and ergodic. Moreover the stationary probability $\Pi_\infty(i) = c^{|m_i|} / \sum_{j \in M} c^{|m_j|}$.*

In [Sin93] it is also shown that for graphs verifying

$$\frac{|M_n(G)|}{|M_{n-1}(G)|} \leq q(n) \quad (3)$$

for a polynomial function q , then taking $c = 2q(n)$, the chain \mathcal{M} converges to an uniform stationary distribution on the subset of perfect matchings. Furthermore in the limit distribution, the probability of getting a perfect matching is bigger than $1/2$. As every dense graph satisfies equation (3), then the class of bipartite dense graphs is a subset of the class of graphs satisfying (3). Notice that the stationary distribution is non-uniform on the set M .

We define a *genetic system* \mathcal{G} over the population of all matchings M that will produce the next generation according to a mating rule grounded in the transitions of \mathcal{M} .

Definition 1 (Mating Rule). *From parents m_l and m_r , sort randomly the edges of m_r . The offspring m_k is the matching resulting of applying the following procedure:*

- (1) With probability $1/2$, $m_k = m_l$.
Otherwise,
 - (2.1) For every edge in $m_r \cap m_l$ with probability $1/(1+c)$ choose that the edge does not belong to m_k .
 - (2.2) For every edge $e = (u, v) \in m_r$ such that u and v are unmatched in m_l , with probability $c/(1+c)$ choose e to be in m_k .

(2.3) For every maximal increasing path in $m_r \cap m_l$, starting in a node unmatched in m_l , and having even length, see Figure 3. Each edge in the path coming from m_r , with probability $1/2$ choose label 1, otherwise choose label 0. Beginning with the first edge in the path, compute the longest prefix formed with edges labelled 1 (if any). Then m_k consists of the edges from m_r in the prefix, and the edges from m_l after the first edge labelled 0.

Given three matchings m_i , m_j and m_k , let $P(i, j, k)$ denote the probability of getting m_k as an offspring of m_i and m_j .

To define a system evolving in time t , start from a given initial generation Π_0 over M at $t = 0$. The generation at time $t + 1$ is obtained from the generation Π_t at time t , by sampling two matchings m_l and m_r according to Π_t , and applying the mating rule to m_l and m_r . The system evolves according to the following dynamical equation,

$$\Pi_{t+1}(k) = \sum_{m_l \in M} \Pi_t(l) \cdot \sum_{m_r \in M} P(l, r, k) \cdot \Pi_t(r) \quad (4)$$

Let us consider a probability distribution Π on the set M of all matchings in G . Let us define a Markov chain $\mathcal{M}(\Pi)$ on the set of states M , and using the mating operation as rule for the transitions. Formally the transitions are defined,

Given a matching $m_l \in M$,

- (1) Sample a matching m_r according to distribution Π ,
- (2) move to the matching m_k defined by the mating of m_l and m_r .

From the way the Markov chain has been defined, it evolves accordingly to the following equation,

$$\Pi_{t+1}(k) = \sum_{m_l \in M} \Pi_t(l) \cdot \sum_{m_r \in M} P(l, r, k) \cdot \Pi(r). \quad (5)$$

Notice that The l, k coefficient $\Pi(l, k)$ in the transition matrix of $\mathcal{M}(\Pi)$ is given by $\Pi(l, k) = \sum_{m_r \in M} P(l, r, k) \cdot \Pi(r)$.

3.1 Convergence

We want to study the conditions under which the genetic system and the Markov chain converge to the limit distribution Π_∞ of \mathcal{M} .

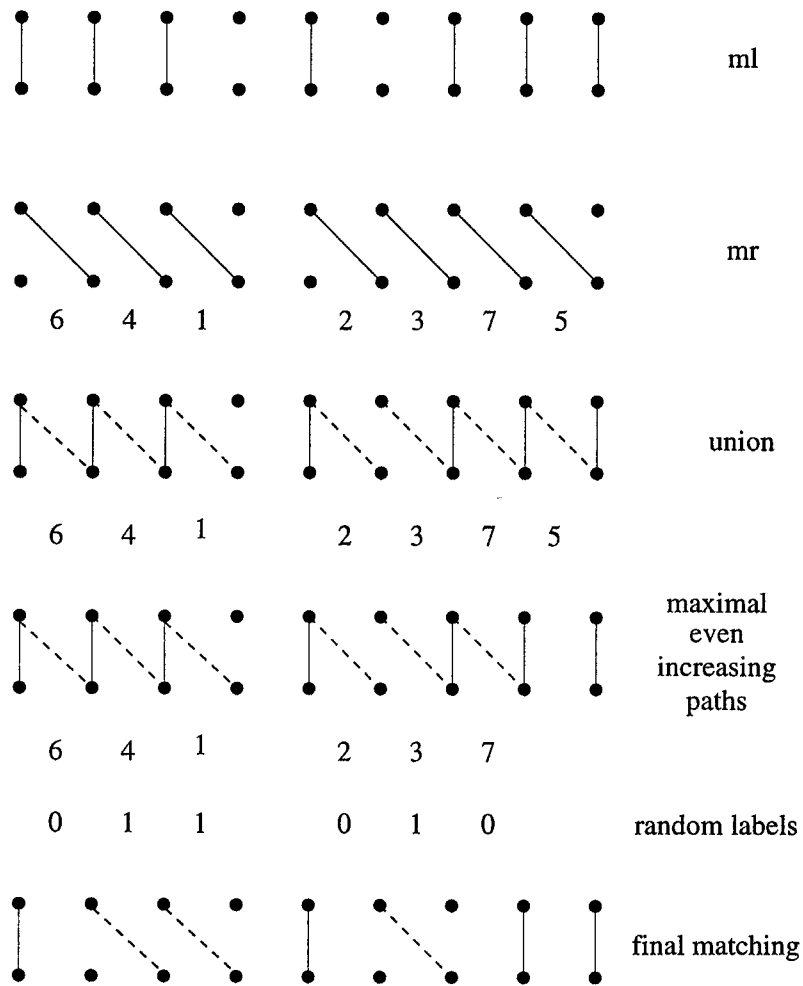


Figure 3: Two matchings and the set of maximal increasing paths.

Lemma 2. *Given a path $\{m_1, \dots, m_r\}$ on the underlying graph of \mathcal{M} , the stationary distribution Π_∞ verifies,*

$$\Pi_\infty(1) \prod_{i=1}^{r-1} p_{i,i+1} = \Pi_\infty(r) \prod_{i=1}^{r-1} p_{i+1,i}.$$

Proof. The time reversibility of \mathcal{M} implies $\Pi_\infty(i) \cdot p_{ij} = p_{ji} \cdot \Pi_\infty(j)$, therefore

$$\begin{aligned} \Pi_\infty(1) \prod_{i=1}^{r-1} p_{i,i+1} &= \Pi_\infty(1) \cdot p_{1,2} \prod_{i=2}^{r-1} p_{i,i+1} \\ &= p_{2,1} \cdot \Pi_\infty(2) \prod_{i=2}^{r-1} p_{i,i+1} \\ &\vdots \\ &= \prod_{i=1}^{s-1} p_{i+1,i} \cdot \Pi_\infty(s) \cdot \prod_{i=s}^{r-1} p_{i,i+1} \\ &\vdots \\ &= \prod_{i=1}^{r-1} p_{i+1,i} \cdot \Pi_\infty(r). \end{aligned}$$

□

Given a matching m_i , we denote by $S(i)$ the set of edge sequences obtained by sorting the edges in m_i . Given three matchings m_l , m_r and m_k , and an element $\vec{r} \in S(r)$ let us denote by $P(l, \vec{r}, k)$ the probability of going from m_l to m_k following a sequence given by \vec{r} in \mathcal{M} , so we have $P(l, r, k) = \sum_{\vec{r} \in S(r)} P(l, \vec{r}, k) / |S(r)|$. Notice that $P(l, r, k) \in [0, 1]$, with $\sum_k P(l, r, k) = 1$. Using lemma 2 and the fact that, when $P(l, \vec{r}, k) \neq 0$ there is a matching m_s of the same size as m_r such that $P(k, \vec{s}, l)$, we get,

Lemma 3. *Given three matchings m_l , m_r and m_k , we have*

$$\Pi_\infty(l) \cdot P(l, r, k) = P(k, r, l) \cdot \Pi_\infty(k).$$

This lemma gives us the property we need to prove convergence of both, $\mathcal{M}(\Pi)$ and \mathcal{G} to the distribution Π_∞ .

Theorem 8. *Given a distribution Π over the set of matchings of a graph $G = (V, E)$. If for every matching m with one edge we have $\Pi(m) > 0$, then $\mathcal{M}(\Pi)$, and \mathcal{G} converge to the limit distribution of \mathcal{M} .*

Proof. In order to prove the convergence of $\mathcal{M}(\Pi)$, we show the time reversibility of $\mathcal{M}(\Pi)$ with respect to the distribution Π_∞ . Recall that the probability of going from m_l to m_k in $\mathcal{M}(\Pi)$ is given by the equation $\Pi(l, k) = \sum_{m_r \in M} P(l, r, k) \cdot \Pi(r)$, therefore using lemma 3 we have

$$\begin{aligned} \Pi_\infty(l) \cdot \Pi(l, k) &= \Pi_\infty(l) \cdot \sum_{m_r \in M} P(l, r, k) \cdot \Pi(r) \\ &= \sum_{m_r \in M} \Pi_\infty(l) \cdot P(l, r, k) \cdot \Pi(r) \\ &= \sum_{m_r \in M} \Pi_\infty(k) \cdot P(k, r, l) \cdot \Pi(r) \\ &= \Pi(k, l) \cdot \Pi_\infty(k) \end{aligned}$$

Let us prove the convergence of the genetic system \mathcal{G} . Recall that the system evolves according to the equation

$$\Pi_{t+1}(k) = \sum_{m_l \in M} \Pi_t(l) \sum_{m_r \in M} P(l, r, k) \cdot \Pi_t(r).$$

Substituting Π_∞ in the previous equation and using again lemma 3 we get,

$$\begin{aligned} \sum_{m_l \in M} \Pi_\infty(l) \cdot \sum_{m_r \in M} P(l, r, k) \cdot \Pi_\infty(r) \\ &= \sum_{m_l \in M} \sum_{m_r \in M} \Pi_\infty(l) \cdot P(l, r, k) \cdot \Pi_\infty(r) \\ &= \sum_{m_l \in M} \sum_{m_r \in M} \Pi_\infty(k) \cdot P(k, r, l) \cdot \Pi_\infty(r) \\ &= \Pi_\infty(k) \sum_{m_r \in M} \Pi_\infty(r) \sum_{m_l \in M} P(k, r, l) \\ &= \Pi_\infty(k) \sum_{m_r \in M} \Pi_\infty(r) = \Pi_\infty(k) \end{aligned}$$

Therefore Π_∞ is a fix point for the system, let us see that it is the unique fix point of the system. Suppose that Δ is another fix point, by the restriction on the initial distribution, Δ must assign positive probability to any matching. Therefore the Markov chain defined using the mating rule and the Δ distribution converges to Π_∞ and Δ therefore $\Delta = \Pi_\infty$. \square

I figure 4 it is given ??????

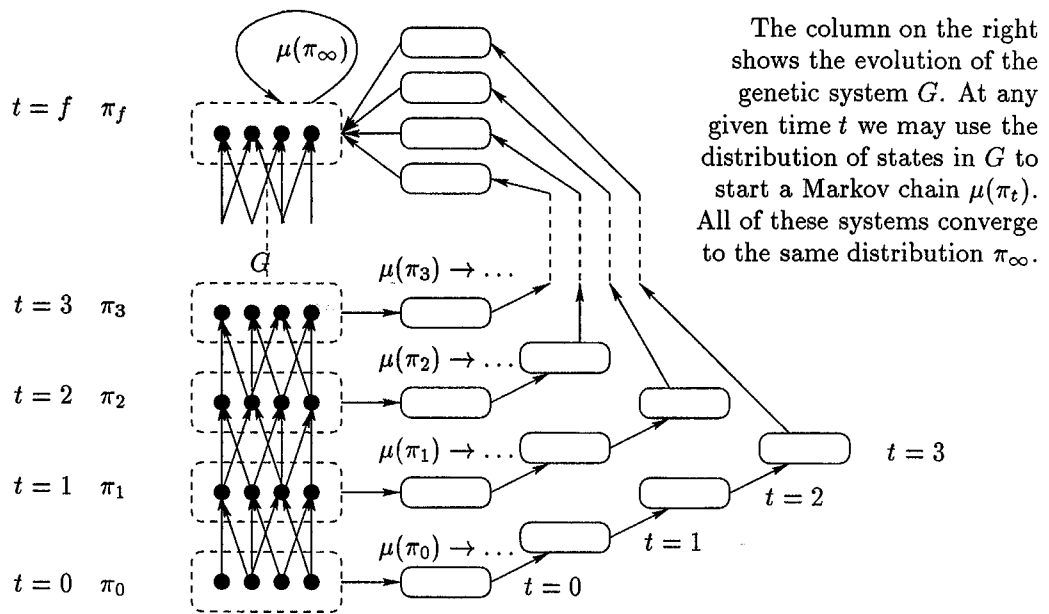


Figure 4: The genetic system G and the Markov chains associated to every distribution.

3.2 RNC computation of the mating rule

Given two matchings m_l and m_r to compute in RNC the mating operation, that gives birth to child m_k , consider the following procedure:

- (1) With probability $1/2$, $m_k = m_l$
Otherwise :
 - (2) In parallel assign an order to the edges in m_r .
 - (3) For every edge in the graph, check if it is in both matchings. If so with probability $1 - 1/(1 + c)$ choose that the edge is in the offspring m_k .
 - (4) For every edge $(u, v) \in m_r$ check whether u and v are unmatched in m_l . If so, with probability $c/(1 + c)$ choose (u, v) to be in m_k .
 - (5) For each free node in m_l , obtain the list of edges in m_r that form a maximal increasing path,
if it has even length.
 - (5.1) Toss the coin and assign labels 0/1 with equal probability to the edges.
 - (5.2) Obtain the first edge a with label 0. All the edges before a that belong to m_r and all the edges after a that belong to m_l form the matching m_k .
- otherwise keep the edges in m_l .

It is easy to implement steps (1) to (4) with a CREW PRAM in $O(\log n)$ steps and μ processors. The data structure we use to represent a matching is an array with $2n$ positions, numbers between 1 and n represent nodes in one bipartition and numbers between $n + 1$ and $2n$ the other one. A value j in position i , $j \neq 0$ means that edge (i, j) is in the matching, when $j = 0$ i is unmatched. That is we keep a doubly linked list of edges given to any edge in the matching both possible orientations. To implement step (5) we add pointers to the data structure. For an edge $e = (y, z)$ in m_l the number of edges in m_r that touches e may be 0, 1, or 2. If this number is 0 we link the edge to itself. When there is only one e' , assuming that $e' = (x', x)$, we link (x', x) with (x, x') . And in the case that there are two edges e', e'' , such that e' is previous to e'' , suppose the $e' = (x', x)$ and $e'' = (y, y')$, we link (x', x) with (x, y) and (x, y) with (y, y') (see figure 5). Finally each node i unmatched by m_l is linked to the corresponding node position in m_r . The additional pointer structure can be computed in $O(1)$ time with $O(n)$

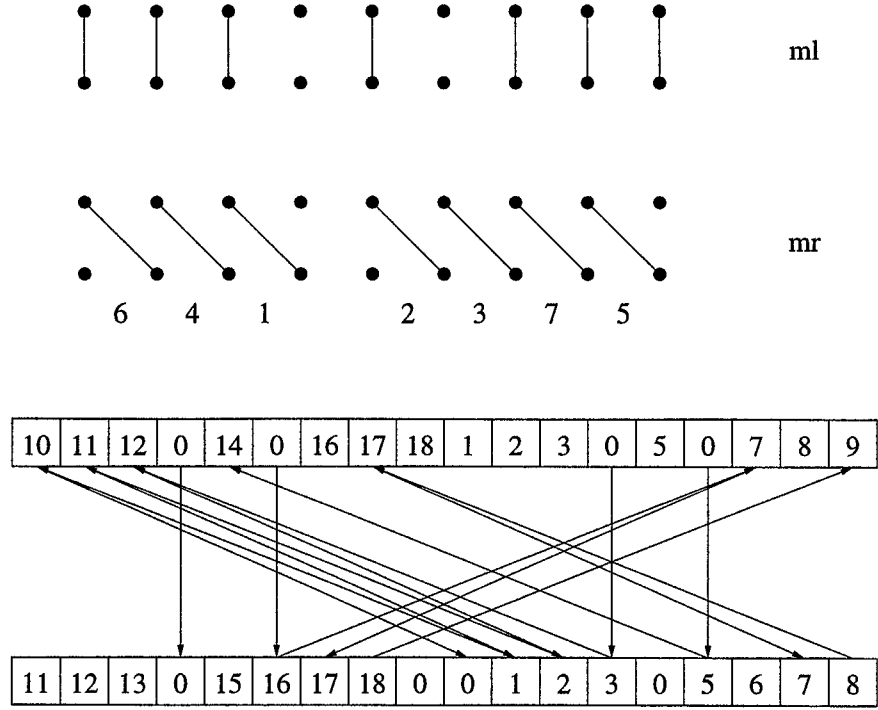


Figure 5: Data structure and links

processors with a CREW PRAM. Starting from the edges leaving free points in m_l , use pointer jumping to obtain the maximal paths and compute their length. Keep those paths which have even length. All this can be done in $O(\log n)$ using $O(n)$ processors in a randomized CREW PRAM, including the label assignment. Finally step (6) can be implemented with the same bounds, using again the pointer jumping technique.

Theorem 9. *The mating that defines the genetic system \mathcal{G} can be computed in RNC.*

4 Local Search and Optimization Problems

One of the characterization of algorithmics in recent time is the use of heuristics greedy type. Those heuristics seems to work quite well in practice for some problems but the theoretical foundations of *why* or *how* they work is an open and difficult topic of research.

Let us recall that given a combinatorial search space S and an objective function $f : S \rightarrow R^+$, a **maximization problem** consists on finding $\sigma^* \in S$ such that $\forall \sigma \in S : f(\sigma) \leq f(\sigma^*)$, ie to find a maximum. A **minimization problem** consists on finding $\sigma^* \in S$ such that $\forall \sigma \in S : f(\sigma) \geq f(\sigma^*)$, ie to find a minimum.

For example the Graph Bisection problem consists in given $G = (V, E)$ with $|V| = n = 2k$, find $V_1, V_2 \in S$ with $|V_1| = |V_2| = n/2$ and $V_1 \cup V_2 = V$ such that $|\{\{u, v\} : u \in V_1, v \in V_2\}|$ is minimum.

Here the search space S is the set of all possible bisections and $f(\sigma)$ is the number of crossing edges. The decisional version of this problem is also NP complete.

A **black-box heuristic** is a randomized search heuristic operating on a connected neighborhood structure H on the vertex set S . Usually, the edge neighbors of a particular state are defined under some measure of distance that is natural to the combinatorial problem under consideration. The term Black-box was coined by Ari Juels in his PhD dissertation [Jue90].

The choice of the neighborhood represents a key decisions in the application of an algorithm and affects much of its performance.

An heuristic that has been used to solve efficiently some difficult optimization problems is **hillclimbing**. It has the following generic structure,

```

function HillClimbing( $S, N, f$ )
  Select initial state  $\sigma \in S$ 
  while movement is possible do
    Randomly select  $\sigma' \in N(\sigma)$ 
    if  $f(\sigma') > f(\sigma)$  then — could also be  $\geq$  —
       $\sigma := \sigma'$ 
    end if
  end while
  return  $\sigma$ 
end

```

The algorithm terminates when it encounters a local maximum (or minimum), i.e. an state σ such that $\forall \sigma' \in N(\sigma) : f(\sigma) \geq f(\sigma')$. A local maximum can be on a “peak” (its objective function value is strictly greater than all its neighbors) or in a “plateau” (its objective function value is greater or equal than all its neighbors. This is why one must decide to choose a $>$ or a \geq sign in the algorithm and be careful in the last case to avoid cycling.

For instance, in [KP92] the authors prove that hillclimbing finds a satisfying truth assignment, if one exists, with high probability.

The problem of the hillclimbing algorithm is that once a local optimum is found the algorithm returns it, but this local optimum can be different

of the global optimum. In order to enable the algorithm to accept downhill moves, the Metropolis algorithm is parametrized by a temperature t and proceeds as follows:

```

function Metropolis( $S, N, f, t$ )
  Select initial state  $\sigma \in S$ 
  while movement is possible do
    Randomly select  $\sigma' \in N(\sigma)$ 
     $\Delta := f(\sigma) - f(\sigma')$ 
    with probability  $\min(1, e^{-\Delta/t})$  do
       $\sigma := \sigma'$ 
    end with
  end while
  return  $\sigma$ 
end

```

Observe that uphill movements will be automatically accepted, whereas downhill movements are accepted randomly in function of the height (Δ) of the movement and the temperature t . With a high temperature the probability of descending is high; with a small temperature, it is low. In the limit, as $t \rightarrow \infty$ Metropolis makes a random walk and as $t \rightarrow 0$ Metropolis becomes the hillclimbing algorithm.

Formally, let $d(\sigma)$ be the degree of σ and $D = \max_{\sigma \in S} \{d(\sigma)\}$. The Metropolis algorithm can be seen as a Markov chain H with transitions defined by

1. Self loop with $\mathbf{Pr} = 1/2$ when $\sigma = \sigma'$.
2. Choose: σ' with probability

$$\mathbf{Pr}(\sigma') = \begin{cases} 1/D & \text{if } \sigma' \in N(\sigma) \\ 1 - d(\sigma)/D & \text{if } \sigma' = \sigma \\ 0 & \text{otherwise.} \end{cases}$$

3. With probability $\min(1, e^{-\Delta/t})$ go to σ' .

Let μ_t be the above chain. Its transition matrix is $P = [p_{\sigma\sigma'}]$ with $p_{\sigma\sigma'} = \mathbf{Pr}(\sigma') \cdot \min(1, e^{-\Delta/t})$. Define $\pi_t(\sigma) = \frac{e^{f(\sigma)/t}}{Z^t}$, where π_t represents the stationary distribution for μ_t . It is straightforward to prove that μ_T is ergodic and reversible. Moreover notice that when $t \rightarrow \infty$, we have $e^{1/t} \rightarrow 1$ and thus π_t is the uniform distribution ($1/|S|$). On the other hand, when $t \rightarrow 0$, π_t becomes more sharply peaked around optimal solutions in S .

Therefore, fixing t , the Metropolis algorithm is just a “sufficiently long” random walk on the Markov chain μ_t .

The Monomer-Dimer procedure described in Section could be seen as a Metropolis algorithm for the problem of finding a matching of maximum cardinality in a graph. Using that chain, it can be proved the following result [SH88],

Theorem 10. *Let $D = |E|$ and $c = e^{1/t}$. Then for graphs such that $M_{n-1}/M_n \leq \text{poly} = c$ and running Metropolis for $\mathcal{O}(n)$ steps, with probability $1 - \frac{1}{n+1}$ a perfect matching is found.*

Another classical example is the Graph Bisection, taht we already mentioned above. Jerrum and Sorkin considered the problem for the following restricted graph model [JS93],

In the model $\mathcal{G}_{4n,p,r}$, a graph G has $4n$ nodes, half colored white and half black. Edges between nodes with the same color are included independently with probability p , while those between nodes of different colors are included with probability r with $r < p$. For sufficiently large values of $p - r$, these graph instances will contain a bisection β^* (the one in which white and black nodes are separated) which is very likely to be the unique minimum. This is refered to as the *planted* bisection.

The value $p - r$ characterises the difficulty. For $p = r = 1/2$, the expected cut of the planted bisection is $n^2/8$. In the case $p = 1/2$ and $r = p - n^{\Delta-2}$ for $3/2 < \Delta < 2$, the expected cut of the planted bisection is $n^2/8 - n^\Delta/4$.

Given a bisection β , we define $b(\beta)$ as its cut. We refer $\sigma(l, r)$ as the operation of swapping two vertices $r \in V_1$ and $l \in V_2$. We define $N(\beta)$ as the set of all possible states obtainable from β by a single move σ . We finally define B as the set of all possible bisections of G . The Metropolis algorithm is given:

```

function Metropolis( $G$ ) is
  Choose  $\beta_0$  uniformly from  $B$ 
  for  $i := 1 \dots t$  do
    Choose randomly  $\beta' \in N(\beta)$ 
     $\Delta C := b(\beta') - b(\beta)$ 
    with probability  $\frac{1}{1+e^{\Delta C/T}}$  do
       $\beta := \beta'$ 
    end with
  end for
  return  $\beta$ 
end

```


Its Markov chain is given by the set of states B and the following transitions:
 1) With probability $1/2$, stay in the same state; 2) Choose $\beta' \in N(\beta)$ and with probability $\frac{1}{1+e^{\Delta C/T}}$ move to it. The transition matrix is $P = [p_{\beta\beta'}]$ with $p_{\beta\beta'} = \Pr(\beta') \cdot \frac{1}{1+e^{\Delta C/T}}$. As we have an ergodic chain, it converges to a stationary distribution π_0 .

Jerrum and Sorkin proved the following result,

Theorem 11. *Let $\epsilon > 0$, $p - r = n^{1-6+2\epsilon}$, $T = n^{5/6+\epsilon}$. Select $G \in \mathcal{G}_{4n,p,r}$. At constant temperature, the metropolis algorithm reaches the unique β^* in $\mathcal{O}(n^{2+\epsilon})$ steps with overwhelming probability.*

Ari Juels [Jue90] demonstrates that these results are, in some measure, also extrapolable to the hillclimbing algorithm. Jerrum and Sorkin [JS93] also prove that T is too high to be effective on small instances. There have been other work on proving formalizing Metropolis algorithms, for instance Nolte and Schrades use similar ideas to the ones developed by Jerrum and Sorkin, to give a kind of Metropolis algorithm for [NS97] 3-colorability, restricted to some particular kind of graphs.

The authors believe that an important and difficult topic of research is the formalization of hillclimbing type algorithms, and their parallel implementation and formalization. For the parallel implementation, non-linear systems could be of help. But they have the inconvenience that little is known about the theory of non-linear dynamic systems. In fact another important and difficult open area of research is to formalize and study the convergence of genetic algorithms of which the systems studied in section 3 are just an oversimplified model.

5 Acknowledgments

We thank Jordi Petit, Xavier Molinero and Marko Rieddel for letting us use some of their LaTeX macros.

References

- [ARV94] S. Arora, Y. Rabani, and U. Vazirani. Simulating quadratic dynamical systems is PSPACE-complete. In *26th ACM Symposium on Theory of Computing*, pages 83–92, 1994.
- [Bro86] A. Broder. How hard is to marry at random. In *18th ACM Symposium on Theory of Computing*, pages 50–58, 1986.

- [Bro89] A. Broder. Generating random spanning trees. In *30th IEEE Symposium on Foundations of Computer Science*, pages 442–447, 1989.
- [Chu96] F.R.K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1996.
- [JS89] M. Jerrum and A. Sinclair. Approximating the permanent. *SIAM Journal on Computing*, 18:1149–1178, 1989.
- [JS93] M. Jerrum and G. Sorkin. Simulated annealing for graph bisection. In *34th IEEE Symposium on Foundations of Computer Science*, pages 94–103, 1993.
- [Jue90] Ari Juels. *Basic in Blax-box Combinatorial Optimization*. Phd thesis, University of California at Berkeley, 1990.
- [JV86] M. Jerrum, L. Valiant, and V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.
- [Kan94] R.. Kannan. Markov chains and polynomial time algorithms. In *35th IEEE Symposium on Foundations of Computer Science*, pages 656–671, 1994.
- [KKL⁺93] N. Karmarkar, R. Karp, R. Lipton, L. Lovasz, and M. Luby. A Monte-Carlo algorithm for estimating the permanent. *SIAM Journal on Complexity*, 22:284–293, 1993.
- [KLM89] R.M.. Karp, M. Luby, and N. Madras. Monte-Carlo approximation algorithms for enumeration problems. *Journal of Algorithms*, 18:429–448, 1989.
- [KP92] E. Koutsupias and X. Papadimitriu. On the greedy algorithm for satisfiability. *Information Processing Letters*, 43:53–55, 1992.
- [Lin92] Torgny Lindvall. *Lectures on the Coupling Method*. Wiley Inter-Science, 1992.
- [Nor97] J.R. Norris. *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics, 1997.
- [NS97] A. Nolte and R. Schrader. Coloring in sublinear time. In R. Burkard and G. Woeginger, editors, *Algorithms-ESA '97*,

- pages 388–401. Lecture Notes in Computer Science, Springer-Verlag, 1997.
- [Pap94] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, Mass., 1994.
 - [Pud94] P. Pudlak. Complexity theory and genetics. In *9th. IEEE Symposium on Structure in Complexity Theory*, pages 383–395. IEEE, 1994.
 - [RRS95] Y. Rabani, Y. Rabinovich, and A. Sinclair. A computational view of population genetics. In *27th ACM Symposium on Theory of Computing*, pages 83–92, 1995.
 - [RSW92] Y. Rabinovich, A. Sinclair, and A. Wigderson. Quadratic dynamical systems. In *33th IEEE Symposium on Foundations of Computer Science*, pages 304–313, 1992.
 - [SH88] G.H. Sasaki and B. Hajek. The time complexity of maximum matching by simulate annealing. *Journal of the ACM*, 35:387–403, 1988.
 - [Sin93] A. Sinclair. *Algorithm for random generation and counting: A Markov chain approach*. Birkhäuser, Boston, 1993.
 - [Ten95] Shang-Hua Teng. Independent sets versus perfect matchings. *Theoretical Computer Science*, pages 1–10, 1995.
 - [Val79a] L.G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.
 - [Val79b] L.G. Valiant. The complexity of enumeration and reliability problems. *SIAM J on Computing*, 8:410–421, 1979.
 - [Vaz91] V. Vazirani. Rapidly mixing Markov chains. In B. Bollobas, editor, *Probabilistic combinatorics and its applications*, pages 99–121. American Mathematical Society, 1991.

Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya

Research Reports – 1997

- LSI-97-1-R “On the Number of Descendants and Ascendants in Random Search Trees”, Conrado Martínez and Helmut Prodinger.
- LSI-97-2-R “On the Epipolar Geometry and Stereo Vision”, Blanca García de Diego.
- LSI-97-3-R “Solving Incidence and Tangency Constraints in 2D”, Núria Mata.
- LSI-97-4-R “*Designer*: A Tool to Design and Model Workflows”, Camilo Ocampo, Pere Botella.
- LSI-97-5-R “OBJECTFLOW: A Modular Workflow Management System”, Camilo Ocampo, Pere Botella.
- LSI-97-6-R “The Extreme Vertices Model (EVM) for Orthogonal Polyhedra”, A. Aguilera and D. Ayala.
- LSI-97-7-R “An Improved Master Theorem for Divide-and-Conquer Recurrences”, Salvador Roura.
- LSI-97-8-R “Randomized Binary Search Trees”, Conrado Martínez and Salvador Roura.
- LSI-97-9-R “Programming Frames for the Efficient Use of Parallel Systems”, Thomas Römke and Jordi Petit i Silvestre.
- LSI-97-10-R “Concurrent Rebalancing of AVL Trees: A Fine-Grained Approach”, Luc Bougé, Joaquim Gabarró, Xavier Messeguer, and Nicolas Schabanel.
- LSI-97-11-R “The VEX-93 Environment as a Hybrid Tool for Developing Knowledge Systems with Different Problem Solving Techniques”, Julio J. Valdés, Ramón Hita, Katia Peón, Dania Hernández, Ada García, Raul Paredes, Yazna García, and Alfredo Rodríguez.
- LSI-97-12-R “Single-Pushout Hypergraph Rewriting through Free Completions”, Ricardo Alberich, Francesc Rosselló, and Gabriel Valiente.
- LSI-97-13-R “Design, implementation and evaluation of ParaDict, a data parallel library for dictionaries”, Joaquim Gabarró and Jordi Petit i Silvestre.
- LSI-97-14-R “NoFun: A Notation to State Non-Functional Specifications at the Product Level”, Xavier Franch.
- LSI-97-15-R “Discontinuities in Recurrent Neural Networks”, Ricard Gavaldà and Hava T. Siegelmann.
- LSI-97-16-R “A Dichotomy Theorem for Learning Quantified Boolean Formulas”, Víctor Dalmau.
- LSI-97-17-R “Rule generation from real data: GAR meets LINNEO+”, David Riaño and Ulises Cortés.

- LSI-97-18-R “Approximating Scheduling Problems in Parallel”, Maria Serna and Fatos Xhafa.
- LSI-97-19-R “Especificación de restricciones de integridad en el sistema ROSES” (written in Spanish), Maria Amélia Pacheco e Silva and Maria Ribera Sancho i Samsó.
- LSI-97-20-R “Octrees Meet Splines”, A. Vinacua, I. Navazo, and P. Brunet.
- LSI-97-21-R “Volume-Based Polyhedra Simplification Based on TG-Maps”, C. Andujar, D. Ayala, and P. Brunet.
- LSI-97-22-R “Refining Logical Characterizations of Advice Complexity Classes”, Albert Atserias and José L. Balcázar.
- LSI-97-23-R “Single-pushout rewriting in categories of spans I: The general setting”, Miquel Monserrat, Francesc Rosselló, Joan Torrens, and Gabriel Valiente.
- LSI-97-24-R “Concurrent Rebalancing on HyperRed-Black trees”, Joaquim Gabarró, Xavier Messeguer and Daniel Riu.
- LSI-97-25-R “Shortcuts: Abstract “Pointers””, J. Marco and X. Franch.
- LSI-97-26-R “The (Parallel) Approximability of Non-Boolean Satisfiability Problems and Restricted Integer Programming”, Maria Serna, Luca Trevisan and Fatos Xhafa.
- LSI-97-27-R “Mean Field Theory of Fluid Neural Networks”, Jordi Delgado and Ricard V. Solé.
- LSI-97-28-R “The Structure of Logarithmic Advice Complexity Classes”, José L. Balcázar and Montserrat Hermo.
- LSI-97-29-R “Structuring the Process of Integrity Maintenance (Extended Version)”, Enric Mayol and Ernest Teniente.
- LSI-97-30-R “Unranking of Combinatorial Structures”, Xavier Molinero Albareda.
- LSI-97-31-R “Parallel Dictionaries with Local Rules on AVL and Brother Trees”, Joaquim Gabarró and Xavier Messeguer.
- LSI-97-32-R “Temporal Features of Class Populations and Attributes in Conceptual Models”, Dolors Costal, Antoni Olivé and Maria-Ribera Sancho.
- LSI-97-33-R “A Unified Approach to Concurrent and Parallel Algorithms on Balanced Data Structures”, Joaquim Gabarró and Xavier Messeguer.
- LSI-97-34-R “A Short Note on Non-Symmetric Semidefinite Programming”, Fatos Xhafa.
- LSI-97-35-R “On Schema and Functional Architectures for Multilevel Secure and Multiuser Model Federated DB Systems”, Elena Rodríguez, Marta Oliva, Fèlix Saltor and Benet Campderrich.
- LSI-97-36-R “Collaboration between Human and Artificial Societies”, Julian Padget, Carles Sierra, Ulises Cortes, Miquel Sánchez i Marrè and Javier Bejar.

- LSI-97-37-R "Combining Constructive and Equational Geometric Constraint Solving Techniques", R. Joan-Arinyo and A. Soto-Riera.
- LSI-97-38-R "Fringe analysis for parallel MacroSplit insertion algorithms in 2-3 trees", R. Baeza-Yates, J. Gabarró and X. Meseguer.
- LSI-97-39-R "Empirical results on long-lived renaming algorithms", Sandra Moral and José Luis Balcázar.
- LSI-97-40-R "Refining Logical Characterizations of Advice Complexity Classes", Albert Atserias and José Luis Balcázar.
- LSI-97-41-R "Approximation Heuristics and Benchmarkings for the MinLA Problem", Jordi Petit i Silvestre.
- LSI-97-42-R "Path Orderings, Quasi-orderings and Termination of Term Rewriting System", Cristina Borralleras and Albert Rubio.
- LSI-97-43-R "Some Dichotomy Theorems on Constant-free Quantified Boolean Formulas", Victor Dalmau Lloret.
- LSI-97-44-R "CAD and the Product Master Model", Christoph M. Hoffmann and Robert Joan-Arinyo.
- LSI-97-45-R "Geometry of Language", Glyn Morrill.
- LSI-97-46-R "Combining Spectral Sequencing with Simulated Annealing for the MinLA Problem: Sequential and Parallel Heuristics", Jordi Petit i Silvestre.
- LSI-97-47-R "DOMAINS, RELATIONS AND RELIGIOUS WARS", Rafael Camps.
- LSI-97-48-R "ATLAS, a platform for distributed graphics applications", Marta Fairen and Alvar Vinacua.
- LSI-97-49-R "Interprocess data transfer in ATLAS, a platform for distributed applications", Marta Fairen and Alvar Vinacua.
- LSI-97-50-R "ATLAS: a platform for transparently developing distributed applications", Marta Fairen and Alvar Vinacua.
- LSI-97-51-R "An algebraic framework for the definition of compositional semantics of Normal Logic Programs", Paqui Lucio, Fernando Orejas y Elvira Pino.
- LSI-97-52-R "The structural events in the ROSES language (Written in Catalan)", Dolors Costal, Anna Roselló and Maria-Ribera Sancho.
- LSI-97-53-R "Generic CSP Techniques for the Job-Shop Problem", Javier Larrosa and Pedro Meseguer.
- LSI-97-54-R "Partial Lazy Forward Checking", Javier Larrosa and Pedro Meseguer.
- LSI-97-55-R "Using Projective Octrees in 3D Reconstruction", Blanca Garcia de Diego and Pere Brunet i Crosa.

LSI-97-56-R "Linear and Non-linear Systems: A survey", Josep Diaz, Maria Serna and Paul Spirakis.

Hardcopies of reports can be ordered from:

Nuria Sánchez
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Campus Nord, Mòdul C6
Jordi Girona Salgado, 1-3
08034 Barcelona, Spain
secrelsi@lsi.upc.es

See also the Department WWW pages, <http://www-lsi.upc.es/www/>


BIBLIOTECA RECTOR GABRIEL FERRATE
Campus Nord