Markov Chains and Random Walks

Jinqiao Yu
LDCSEE
West Virginia University,
Morgantown, WV
{jyu@csee.wvu.edu}

Let $G = (V, E)$ be a connected, undirected graph with n vertices and m edges. For a vertex $v \in V$, $\Gamma(v)$ denotes the set of neighbors of v in G. A *random walk* on G is the following process, which occurs in a sequence of discrete *steps:* starting at a vertex v_0, we proceed at the first step to a random edge incident on v_0 and walking along it to a vertex v_1, and so on. "Random chosen neighbor" will mean a neighbor chosen uniformly at random; the choice at each step is independent of all previous choices.

Two typical questions about *random walk* in a complete graph K_n:

Let u and v be two vertices in K_n.
1.Starting from vertex u, what's the expected number of steps of first reaching v?
2.Starting from vertex u, what's the expected number of steps to visit every vertex in the graph?

Answers:
1. The expected number of steps starting from u to first reach v is $n - 1$.
Solution: Starting from u, at any step the probability of first reaching a specific vertex v is $\frac{1}{n-1}$.
Let X be the number of steps of first reaching v from u. Then the distribution of X is a geometric distribution with $p = \frac{1}{n-1}$. Therefore expected number of steps of first reaching v from u=$E(X) = \frac{1}{p} = n - 1$.
2. The expected number of steps starting from u to visit all the vertices in K_n is $(n-1)H_{n-1}$, where $H_{n-1} = \sum_{j=1}^{n-1} 1/j$ is the Harmonic number.
Solution: Let X be a random variable defined to be the number of steps required to visit all vertices in K_n. Let $C_1, C_2, ..., C_X$ denote the sequence of steps, where C_i denotes the vertex visited at step i. Call the ith step a success if the vertex was not visited in any of the first $i - 1$ steps.
We divide the sequence into *epochs*, where epoch i begins with the step following the ith success and ends with the step on which we visit the $(i + 1)st$ vertex. Define the random variable X_i, for $0 \leq i \leq n - 2$, to be the number of steps in the ith epoch, so that

$$X = \sum_{i=0}^{n-2} X_i.$$

Further, let p_i denote the probability of success on any step of the ith epoch. This is the probability of visiting one of the $n - 1 - i$ remaining vertices and so,

$$p_i = \frac{n - i - 1}{n - 1}.$$

The random variable X_i is geometrically distributed with parameter p_i. Thus, the expected value of X_i is $1/p_i$. By linearity of expectation,

$$E[X] = E[\sum_{i=0}^{n-2} X_i] = \sum_{i=0}^{n-2} E[X_i] = \sum_{i=0}^{n-2} \frac{n-1}{n-i-1} = (n-1)\sum_{i=1}^{n-1} \frac{1}{i} = (n-1)H_{n-1}.$$

1 2-SAT PROBLEM

Recall that the $k-SAT$ problem is the special case of the SAT problem in which each clause contains exactly k literals. We seek an assignment of Boolean values to the variables such that all the clauses are satisfied, or an assurance that no such assignment exists. While the $k-SAT$ problem is NP-hard for $k \geq 3$, it is solvable in polynomial time for $k = 1$ or $k = 2$. Below is a polynomial-time algorithm for $2-SAT$ problem ($k = 2$).

Function MON2SAT-ALGORITHM

1: Start with a random assignment.
2: While there is an unsatisfied clause(both of the two literal values are false).
a.Choose one of its literals uniformly at random and flip it.
3: After r repetitions declare that the formula is probably unsatisfiable.

Algorithm 1: MON2SAT

For this algorithm, if the $2-SAT$ problem is truly unsatisfiable, then it will definitely says so. But if the algorithm says that the $2-SAT$ problem is unsatisfiable, it is probably incorrect. (The $2-SAT$ problem is actually satisfiable, but the algorithm failed to find a satisfying assignment.) Therefore it is a RP algorithm. From the following theorem, we will know that it's expected running time is polynomial. Consequently, the above algorithm is a Monte-Carlo algorithm with one-sided error.

***Theorem:* 1.1.** *If $r = 2n^2$, where r is the total number of coin tosses, the probability that a satisfying assignment will be found by Algorithm MON2SAT for a satisfied formula $\geq \frac{1}{2}$.*

Proof:
Let T represents a satisfying assignment. Let $t(i)$=expected number of steps for the algorithm to reach a satisfying assignments given that our current assignment differs from T in exactly i literals.
We want to know what is $t(n)$ =?
It is easy to see

$$t(0) = 0.$$
$$t(n) = 1 + t(n-1).$$

At each iteration, we complement the current value of one of the literals of some unsatisfied clause, so that the number of literals having correct values either is increased by one or decreased by one. Since in an unsatisfied clause, at least one of the two literals has an incorrect value and we randomly choose one of the two literals, therefore with probability at least $\frac{1}{2}$ we increase by one the number of literals having correct values. With probability at most $\frac{1}{2}$, we increase by one the number of literals having incorrect values. Since $t(i-1) \leq t(i+1)$, for $t(i)$, we have $t(i) \leq \frac{1}{2}t(i+1) + \frac{1}{2}t(i-1) + 1$.
Now, let $X(i) = \frac{1}{2}(X(i-1) + X(i+1)) + 1$, then $X(i) \geq t(i)$.

$$X(0) = 0.$$
$$X(n) = 1 + X(n-1).$$

Solving for $X(i)$, we have $X(n) = n^2$. Since $t(i) \leq X(i)$, we have that $t(n) \leq n^2$. Therefore the expected number of steps for the MON2SAT algorithm to find a satisfying assignment is $O(n^2)$. Using Markov's Inequality, it is easy to

see $\mathbf{Pr}[Y > 2n^2] \leq \frac{1}{2}$, *where* Y *is the number of steps for the algorithm to reach a satisfying assignment. Therefore if* $r = 2n^2$, $\mathbf{Pr}[$*a satisfying assignment will be found by Algorithm MON2SAT for a satisfied formula*$] \geq \frac{1}{2}$. □

2 Markov Chains

***Definition:* 2.1.** *A Markov Chain* M *is a discrete-time stochastic process defined over a set* S *of states in terms of a matrix* P *of transition probabilities.The set* s *is either finite or countably infinite. The transition probability matrix* P *has one row and one column for each state in* S*. The entry* P_{ij} *is the probability that the next state will be* j*, given that the current state is* i*. Thus, for all* $i, j \in S$*, we have* $0 \leq P_{ij} \leq 1$*, and* $\sum_j P_{ij} = 1$.

As the possion distribution, an important property of a Markov Chain is the *memorylessness property:* the future behavior of a Markov chain depends only on its current state, and not on how it arrived at the present state. The *memorylessness* property can be stated more formally as follows:

$$Pr[X_{i+1} = j | X_0 = i_0, X_1 = i_1, ..., X_t = i] = Pr[X_{t+1} = j | X_t = i] = P_{ij}.$$

A Markov Chain need not have a prespecified initial state; in general, its initial state X_0 is permitted to be chosen according to some probability distribution over S.
For $i, j \in S$, the t-step transition probability is defined as

$$P_{ij}^{(t)} = Pr[X_t = j | X_0 = i].$$

Given an initial state $X_0 = i$, the probability that the first transition into state j occurs at time t is denoted by $r_{ij}^{(t)}$ and given by

$$r_{ij}^{(t)} = Pr[X_t = j, and\ for\ 1 \leq S \leq t-1, X_s \neq j | X_0 = i].$$

For $X_0 = i$, the probability that there is a visit to j, at some t 0 is given by

$$f_{ij} = \sum_{t>0} r_{ij}^{(t)}.$$

The expected number of steps to reach state j from i is denoted by

$$h_{ij} = \sum_{t>0} t r_{ij}^{(t)}.$$

If $fij < 1$, we have $h_{ij} = \infty$.

***Definition:* 2.2.** *If* $f_{ii} < 1$*,* i *is called transient. If* $f_{ii} = 1$*,* i *is called persistent. A persistent state can be null or non-null, depending on whether* $h_{ii} = \infty$ *or not.*

***Definition:* 2.3.** *A strong component of a directed graph* G *is a maximal subgraph* C *of* G *such that for any pair of vertices* i *and* j *in the vertex set of* C*, there is a directed path from* i *to* j*, as well as a directed path from* j *to* i.

We restrict our attention to finite Markov chains, i.e., Markov chains whose states are finite in number. We claim that every state in such a Markov chain is either transient or non-null persistent. We define the underlying directed graph of a Markov chain as follows: there is one vertex in the graph for each state of the Markov chain; and there is an edge directed from vertex i to vertex j if and only if $p_{ij} > 0$.

***Definition:* 2.4.** *A strong component C is said to be a final strong component* if there is no edge going from a vertex in C to a vertex not in C.

***Theorem:* 2.1.** *A state is persistent if and only if it lies in a final strong component.*

Proof: *For any state i, it must lies in a strong component C. If i is persistent, then we say that the strong component C must be final. Otherwise, there is an edge going from a vertex from C to a vertex not in C, which means there is no directed path from the vertex back to C. Therefore, the probability for state i going out C and can not go back to state i is non-zero. It conflicts with the assumption that the state i is persistent.*

In a finite Markov chain, starting from any vertex in a strong component C, there is a non-zero probability of reaching any other vertex in the same strong component in a finite number of steps. If C is a final strong component, this probability is 1 since the Markov chain can never leave the component C once it enters it. It follows that any state lies in a final strong component must be persistent. □

***Definition:* 2.5.** *A Markov chain is said to be irreducible, if its underlying graph consists of a single strong final component.*

***Definition:* 2.6.** *Define $q^{(t)} = (q_1^{(t)}, q_2^{(t)}, ..., q_n^{(t)})$, the state probability vector (also called the distribution of the chain at time t), to be the row vector whose ith component is the probability that the chain is in state i at time t.*

It is easy to check that $q^{(t+1)} = q^{(t)}p$, so we have by induction that $q^{(t)} = q^{(0)}p^{(t)}$. It follows that a Markov chain's behavior for all time is specified by its initial distribution $q^{(0)}$and its transition matrix P.

***Definition:* 2.7.** *A stationary distribution is a probability distribution π, such that $\pi = \pi p$.*

Intuitively, if the Markov chain is in the stationary distribution at step t, it remains in the stationary distribution at step $t + 1$. Thus the stationary distribution is thought of as a description of the steady-state behavior of the Markov chain.

***Definition:* 2.8.** *The periodicity of a state i is d, if $P_{ii}^{(n)} = 0$ when $d \nmid n$ and d is the largest such integer.*

***Definition:* 2.9.** *A state is said to be periodic if it has periodicity greater than 1, and is said to be aperiodic otherwise. A Markov chain in which every state is aperiodic is known as an aperiodic Markov chain.*

***Definition:* 2.10.** *An ergodic state is one that is aperiodic and non-null persistent.*

***Definition:* 2.11.** *An ergodic Markov chain is one in which all states are ergodic.*

The following is a fundamental theorem of Markov chain.

***Theorem:* 2.2.** *Any irreducible, finite, and aperiodic Markov chain has the following properties.*

1. *All states are ergodic.*
2. *There is a unique stationary distribution π such that, for $1 \leq i \leq n$, $\pi_i > 0$.*
3. *For $1 \leq i \leq n$, $f_{ii} = 1$, and $h_{ii} = 1/\pi_i$.*
4. *Let $N(i, t)$ be the number of times the Markov chain visits state i in t steps. Then*

$$\lim_{t\to\infty} \frac{N(i,t)}{t} = \pi_i.$$

3 Random Walks on Graphs

Let $G = (V, E)$ be a connected, non-bipartite, undirected graph where $|V| = n$ and $|E| = m$. It induces a Markov chain M_G as follows: the states of the M_G are the vertices of G, and for any two vertices $u, v \in V$,

$$P_{uv} = \begin{cases} \frac{1}{d(u)} \; if \; (u,v) \in E \\ 0 \quad otherwise, \end{cases}$$

where $d(u)$ is the degree of vertex u.

***Lemma:* 3.1.** *The periodicity of any state in G, is the gcd of the length of all closed walks in G.*

Since G is undirected, it has a circle of length 2 that traverse the same edge twice in succession. Further, since G is non-bipartite it has odd cycles that give closed walks of odd length. It follows that the *gcd* of the closed walks is 1, and hence M_G is aperiodic. Noting that G is finite, Theorem (2.2) implies that M_G has a unique stationary distribution π.

***Lemma:* 3.2.** *For all $v \in V$, $\pi_v = d(v)/2m$.*

Proof: *Let $[\pi p]_v$ denote the component corresponding to vertex v.*

$$\begin{aligned} \pi_v &= [\pi p]_v \\ &= \sum_u \pi_u p_{uv} \\ &= \sum_{(u,v)\in E} \frac{d(u)}{C} \times \frac{1}{d(u)} \\ &= \frac{d(v)}{C.} \end{aligned}$$

Therefore the system admits the solution $\pi_u = d(u)/C$, where C is a constant. Since $\sum_{v\in V} \pi_v = 1$ and $\sum_{v\in V} d(v) = 2m$, we have $C = 2m$. Therefore the system admits the solution $\pi_v = d(v)/2m$ which, by Theorem (2.2), is the unique stationary distribution. □

As a direct consequence of Theorem (2.2) and Lemma (3.2), we obtain the following lemma.

***Lemma:* 3.3.** *For all $v \in V$, $h_{vv} = 1/\pi_v = 2m/d(v)$.*

***Definition:* 3.1.** *The hitting time h_{uv} (sometimes called the mean first passage time) is the expected number of steps in a random walk that starts at u and ends upon first reaching v.*

***Definition:* 3.2.** *We define C_{uv}, the commute time* between u and v, to be $C_{uv} = h_{uv} + h_{vu} = C_{vu}$. This is the expected time for a random walk starting at u to return to u after at least one visit to v.

***Definition:* 3.3.** *Let $C_u(G)$ denote the expected length of a walk that starts at u and ends upon visiting every vertex in G at least once. The cover time of G, denoted $C(G)$, is defined by $C(G) = max_u C_u(G)$.*

***Lemma:* 3.4.** *For any edge $(u, v) \in E$, $h_{uv} + h_{vu} \leq 2m$.*

Proof: *Replace edges of G by a pair of oppositely directed edges, the directed edges form the state space. There are $2m$ states in this new Markov chain. The transition matrix Q for this Markov chain has non-zero entry*

$$Q_{(u,v),(v,w)} = P_{vw} = 1/d(v),$$

corresponding to an edge (v, w). This matrix is doublystochastic, meaning that not only do the rows sum to one(as in every Markov chain), but the columns sum to one as well. To see this, fix a (directed) edge (v, w) and observe that the column sum corresponding to this state is given by

$$\begin{aligned}\sum_{x\in V, y\in T(x)} Q_{(x,y),(v,w)} &= \sum_{u\in T(v)} Q_{(u,v),(v,w)} \\ &= \sum_{u\in T(v)} P_{vw} \\ &= d(v) \times \frac{1}{d(v)} \\ &= 1.\end{aligned}$$

From the result of Problem 6.6 in the textbook, we have the following results.

1. For any stochastic matrix P, $\exists\pi$, such that $\pi p = \pi$ and $\sum_i \pi_i = 1$.

2. If P is a doubly stochastic matrix, π is necessarily the uniform distribution.

Therefore it follows that the uniform distribution π^Q on the edges is stationary for this Markov chain. Since

$$\pi^Q = [\frac{1}{2m}, \frac{1}{2m}, ..., \frac{1}{2m}].$$

By part (3) of Theorem (2.2), we can conclude that the expected time between successive traversals of the directed edge (v, u) is $2m$.

Consider now $h_{uv} + h_{vu}$, and interpret this as the expected time for a walk starting from vertex u to visit vertex v and return to u. Conditioned on the event that the initial event that the initial entry into u was via the directed edge (v, u), we conclude that the expected time to go from there to v and then to u along (v, u) is $2m$. The memorylessness property of a Markov chain now allows us to remove the conditioning:since the sequence of transitions from u onward is independent of the fact that we arrived at u along (v, u) at the start of the commute, the expected time back to u is at most $2m$. □

4 Electrical Networks

Resistive network is a graph with resistors on the edges.

Kirchoff's law: current is like water flow. For any node, sum entering equal sum leaving.

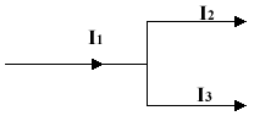

Figure 1: Kirchoff's Law

In figure 1,the arrows represent the direction of current flow, the junction is where the wires meet. I_1 is flowing into the junction whereas I_2 and I_3 are flowing out. If I_1 was 20 amp and I_3 was 5 amp then I_2 would be 15 amp, as $I_1 = I_2 + I_3$. Kirchoff's voltage law states that the sum of voltage drops around a closed circuit is equal to zero. This can also be expressed as the sum of voltage drops around a closed circuit is equal to the sum of voltage sources

Ohm's law: V=IR. (V = voltage difference across resistor).

where V is the Voltage measured in volts, I is the Current measured in amperes, R is the resistance measured in

Ohms.

Figure 2: Olm's Law

In this simple circuit(Figure 2) there is a current of 12 amps (12A) and a resistive load of 1 Ohm (1W). Using the ohm's law from above we determine the Voltage:

$$V = 12 \times 1 = 12\ Volts\ (12V).$$

We have two addition formulas for resistors:
Resistors in series: $R = R_1 + R_2$.
Resistors in parallel: $\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2}$.

Given an undirected graph G, let $N(G)$ be the electrical network defined as follows: it has a node for each vertex in V; for every edge in E, it has a one ohm resistance between the corresponding nodes in $N(G)$. For two vertices $u, v \in V$, R_{uv} denotes the effective resistance between the corresponding nodes in $N(G)$. The following theorem establishes a close relation between commute time for the simple random walk on G and effective resistances in the electrical network $N(G)$.

***Theorem:* 4.1.** *For any two vertices u and v in G, the commute time $C_{uv} = 2mR_{uv}$.*

Proof: *For any vertex u in G. Let $\Gamma(u)$ denote the set of vertices in V that are adjacent to u. Let d(u)=$|\Gamma(u)|$. Let ϕ_{uv} denote the voltage at u in N(G) with respect to v, if d(x) amperes of current are injected into each node $x \in V$, and 2m amperes are removed from v. We claim that for any $u \in V$, $h_{uv} = \phi_{uv}$.*
Using Kirchhoff's Law and Ohm's law, we have

$$\phi_{uv} - \phi_{wv} = \phi_{uw} = 1.$$

Therefore

$$d(u) = \sum_{w \in \Gamma(u)} (\phi_{uv} - \phi_{wv}).$$

By the definition of expectation, for all $u \in V \backslash \{v\}$,

$$h_{uv} = \sum_{w \in \Gamma(u)} \frac{1}{d(u)}(1 + h_{uv}).$$

The above two equations are both linear with unique solutions; furthermore, they are identical if we identify ϕ_{uv} with h_{uv}. This proves $h_{uv} = \phi_{uv}$.To complete the proof of the theorem, we note that h_{uv} is the voltage ϕ_{vu} at v in $N(G)$ measured with respect to v and u, when currents are injected into all nodes and $2mA$ removed from u. Changing signs, ϕ_{vu} is now the voltage at u relative to v when current is injected at u, and removed from all other nodes. Since resistive networks are linear, we can determine C_{uv} by super-posing the networks on which ϕ_{uv} and ϕ_{vu} are measured. Currents at all nodes except u and v cancel, resulting in C_{uv} being the voltage between u and v where $\sum_{w \in V} d(w) = 2m$ amperes are injected into u and removed from v, which yields the theorem by Ohm's law. □

***Lemma:* 4.1.** *The effective resistance between any two nodes u and v is at most the length of the shortest path between them in G.*

Proof: *The effective resistance between any two nodes u and v is the voltage difference between u and v when one ampere is injected into u and removed from v. Let P_1 be the shortest path between u and v. If P_1 is the only path between u and v, then since $V = IR$ and $R = \sum_{i \in P_1} R_i$, we have ϕ_{uv} equals to the length of the shortest path. If there exists a second path p_2. Then the length of $p_2 \geqslant$ the length of p_1. Therefore the resistance of p_2 is greater than the resistance of p_1. Since $\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2}$, we know the resistance is less than the resistance on the shortest path. Therefore the effective resistance between u and v is less the one when there exists only one path between u and v. Hence the effective resistance between any two nodes is at most the length of the shortest path between them.* □

***Definition:* 4.1.** *Diameter of G: the maximum of all of lengths of the shortest path between any 2 vertices in G.*

Obviously, for any n-vertex graph G, we have the diameter of G is less than $n-1$. It is easy to see that G can have at most $n^2/2$ vertices. By Theorem (4.1), we have

$$C_{uv} \leq 2 \times \frac{n^2}{2}(n-1) < n^3.$$

.

***Corollary:* 4.1.** *In any n-vertex graph, and for all vertices u and v,*

$$C_{uv} < n^3.$$

5 Cover Time

***Theorem:* 5.1.**

$$C(G) \leq 2m(n-1).$$

Proof: *Let T be any spanning tree, T can be traversed, such that each edge is visited exactly once in each direction. V_0 be any vertex on G. Let's denote the sequence of the traversal of the tree.*

$$v_0, v_1, \ldots, v_{2n-2} = v_0.$$

$$C_{v_0}(G) \leq \sum_{j=0}^{2n-3} h_{v_j, v_j+1} = \sum_{(u,w) \in T} C_{uw}.$$

Since $(u, w) \in E$, by Lemma (3.4), $C_{uw} \leq 2m$.
It follows that

$$C_{v_0}(G) \leq 2m(n-1).$$

Since v_0 can be any vertex in G, we have

$$C(G) \leq 2m(n-1).$$

□

Let $R(G) = max_{u,v \in V} R_{uv}$; we call R the resistance of G. The resistance of a graph characterizes its cover time fairly tightly.

***Theorem:* 5.2.** *$mR(G) \leq C(G) \leq 2\ell^3 R(G) \ln n + n$.*

Proof: *Notice that there exists vertices u, v such that $R(G)$=R_{uv}. We know $C_{uv} = h_{uv} + h_{vu}$. By Theorem (4.1)), we know $C_{uv} = 2mR_{uv}$. It follows that $mR(G) = m \times \frac{C_{uv}}{2m} = \frac{C_{uv}}{2}$. Since $\frac{C_{uv}}{2} \leq max\{h_{uv}, h_{vu}\}$, we have $mR(G) \leq C(G)$.*

Divide the random walk of length $2\ell^3 mR(G) \ln n$ into $\ln n$ epochs each of length $2\ell^3 mR(G)$. For any vertex v, the hitting time h_{uv} is at most $2mR(G)$, regardless of the vertex u at which an epoch starts. By the Markov inequality, the probability that v is not visited during any single epoch is at most $1/\ell^3$. Therefore the probability that any vertex is not visited within $2\ell^3 mR(G) \ln n$ steps is at most $1/n^2$. When this happens(there is a vertex that has not been visited within $2\ell^3 mR(G) \ln n$ steps , we continue the walk until all vertices are visited, and n^3 steps suffice for this(by Corollary (4.1)). Thus the expected total time is at most

$$2\ell^3 mR(G) \ln n + (1/n^2)n^3 = 2\ell^3 mR(G) \ln n + n.$$

□

6 Graph Connectivity

We are now ready for our first algorithmic application of random walks. Two vertices in an undirected graph G are said to be *connected* if there exists a path between them. A *connected component* of G is a (maximal) subset of vertices in which every pair of vertices is connected.

The *undirected s-t connectivity* (USTCON) problem is the following:given an undirected graph G and two vertices s and t in G, decide whether s and t are in the same connected component.

A probabilistic log-space Turing machine for a language L is a probabilistic Turing machine using space $O(\log n)$ on instances of size n, and running in time polynomial in n. We say that a language(equivalently, a decision problem) A is in RLP if there exists a probabilistic log-space Turing machine M such that on any input x,

$$Pr[M \text{ accepts } x] \begin{cases} \geq \frac{1}{2} & x \in A \\ 0 & x \notin A. \end{cases}$$

***Theorem:* 6.1.** *$USTCON \in RLP$.*

Proof: *The log-space probabilistic Turing machine simulates a random walk of length $2n^3$ through the input graph, starting from s. If it encounters t, it then says YES; otherwise, it outputs NO. Clearly the machine will never say YES on an graph of USTCON in which s and t are not in the same component. What's the probability that it outputs NO when it should have said YES.*

By Theorem (4.1), $h_{st} \leq n^3$. By the Markov inequality, if t is in the same component of G as s, the probability that it is not visited in a random walk of $2n^3$ steps starting from s is at most $1/2$. The Turing machine uses its workspace to count up to $2n^3$, and to keep track of its position in the graph during the walk; both of these require only space $O(\log n)$.

□

We now consider a specific class of non-uniform, deterministic log-space algorithms for USTCON known as *universal traversal sequences.* We focus on n-vertex graphs that are *regular of degree d.* Such a graph is said to be *labeled* if, at each vertex in the graph, each of the d edges incident on that vertex has a unique(integer) label in $\{1, ..., d\}$. There is no requirement that an edge receive the same label at both end-points.

Any sequence of symbols $\sigma = (\sigma_1, \sigma_2, ...)$ from $\{1, ..., d\}$ together with a starting vertex v in a labeled graph describes a walk through the graph.

A sequence σ is said to *traverse* a labeled graph G if the walk it prescribes visits every vertex of G regardless of the starting vertex. A sequence σ is said to be universal traversal sequence for a class of labeled graphs if it traverses every graph in the class, and for every starting vertex.

Let $\mathcal{G}$ be a family of connected labeled $d-$regular graphs on n vertices. Each member of each graph counts as a distinct member of $\mathcal{G}$. Let $U(\mathcal{G})$ denote the length of the shortest universal traversal sequence for all the labeled graphs in $\mathcal{G}$. Let $R(\mathcal{G})$ denote the maximum resistance between any pair of vertices in any graph in $\mathcal{G}$.

***Theorem:* 6.2.** $U(\mathcal{G}) \leq 5mR(\mathcal{G})log_2(n|\mathcal{G}|)$.

Proof: *Given a labeled graph $G \in \mathcal{G}$, let v be a vertex of G. Consider a random walk of length $5mR(\mathcal{G})log_2(n|\mathcal{G}|)$, divided into $log_2(n|\mathcal{G}|)$ "epochs" each of length $5mR(\mathcal{G})$. The probability that the walk fails to visit v in any epoch is at most 2/5 by Theorem (4.1) and Markov's inequality, regardless of the vertex of $\mathcal{G}$ at which the epoch began. The probability that v is not visited during any of the $log_2(n|\mathcal{G}|)$ epochs is thus at most $(2/5)^{log(n|\mathcal{G}|)}$, which can be written into $(n|\mathcal{G}|)^{-\rfloor}$ for a value of $c > 1$. Summing this probability over all n choices of the vertex v and all $|\mathcal{G}|$ choices of the labeled graph G, the probability that the random walk (sequence) fails to be universal is less than one. Thus there is a sequence of this length that is universal for the class $\mathcal{G}$.* □

Let $U(d, n)$ denote the length of the shortest universal traversal sequence for connected, labeled, n-vertex, d-regular graphs.

***Exercise:* 6.1.** *Show that the number of labeled n-vertex graphs that are d-regular is $(nd)^{O(nd)}$.*

Proof. First choose any vertex u, there are n different ways to label it. There are $d!$ different ways to label the edges incident on it. Therefore there are $n \times d!$ different ways to label u and its incident edges. For a second randomly chosen vertex, say v, there are $n-1$ different ways to label it and $d!$ different ways to label the edges incident on it. Therefore there are $(n-1) \times d!$ different ways to label v and its incident edges. Similarly, for the third vertex, there are $(n-2) \times d!$ different ways to label it and its associated edges. Consequently, there are total $(n \times d!) \times ((n-1) \times d!)... \times (1 \times d!) = (n!)(d!)$ different ways to label the graph. Therefore, let X denote the number of labelled $n-$vertex graphs that are d-regular. $X = (n!)(d!)$.

$$\begin{aligned} X &= (n!d!) \\ &= nd \times (n-1)(d-1) \times (n-2)(d-2) \times ... \\ &\leq (nd)^{(nd)} \\ &= (nd)^{O(nd)} \end{aligned}$$

□

***Corollary:* 6.1.** $U(d, n) = O(n^3 d \log n)$.

Proof: *The diameter of every connected n-vertex, d-regular graph is $O(n/d)$ and so, therefore, is its resistance. The number of edges $m = nd/2$. The result now follows from Exercise (6.1) and Theorem (6.2).* □

For directed graphs, we can't use the same techniques used in undirected graphs, because a random walk may trapped at a vertex which does have edges going out. In this case, we modify the technique by jump back from where we got stuck.

As before, let the edges leaving a vertex v be labeled $1, 2, ..., d(v)$. Thus any path in the graph can be associated with a string whose symbols are drawn from $1, 2, ..., n-1$, as in the discussion of universal traversal sequences. If we could begin at s and enumerate the walks corresponding to all such strings of length $n-1$, we would be assured of discovering a path from s to t if one existed. The number of such strings being of the order n^n, we would require $\Omega(n \log n)$ space to maintain a counter that could index these strings. Since we only wish to use $O(n \log n)$ space, we use randomization to achieve this reduction in space.

The algorithm consists of repeatedly executing the following two steps until either step results in termination.

Function A Monte Carlo Algorithm for STCON in directed graphs

1: Starting at s, simulate a random walk of $n-1$ steps. Each step consists of choosing an edge leaving the current vertex uniformly at random. If t is reached, output YES and stop. If the walk reaches a vertex with no outgoing edge, or a vertex other than t after $n-1$ steps, return to s. This step can be implemented using $O(\log n)$ bits of memory.

2: Flip $\log n^n$ unbiased coins. If they all come up HEADS, halt and output NO. This can be implemented by a counter that keeps track of the number of coins that have been flipped. The number of bits required in this counter is $\log(\log n^n)$, which is $O(\log n)$, as required.

Algorithm 2: A Monte Carlo Algorithm for STCON in directed graphs

Since the number of distinct walks from s is at most n^n, the probability of discovering an $s-t$ path on a trial(in Step 1) is at least n^{-n}. The probability of terminating in Step 2 on a trial is the probability that all the coins come up HEADS, and this is n^{-n}. Thus on each trial, the algorithm terminates successfully with probability at least n^{-n}, and erroneously with probability at most $(1-n^{-n})n^{-n} \leq n^{-n}$. Let p_w denote the probability of outputting YES on termination; then we have

$$p_w \leq n^{-n} + (1-2n^{-n})p_w,$$

where the first term on the right-hand side denotes the probability of succeeding on the very first trial, while the second term denotes success thereafter. Solving, we have $p_w \geq 1/2$.

***Theorem:* 6.3.** *The above algorithm will, given an instance of STCON,*

1. *Always output NO if there is no path from s to t.*
2. *Output YES with probability at least 1/2 if there is a path from s to t.*

The algorithm uses space $O(\log n)$.