Randomness and Computation
or, “Randomized Algorithms”

Mary Cryan
School of Informatics
University of Edinburgh

RC (2016/17) – Lectures 11 and 12 – slide 1

Hamiltonian cycles

Definition
Given an undirected graph $G = (V, E)$ with $V = [n]$, an Hamiltonian circuit (HC) of $G$ is a permutation $\pi$ of the vertex set $[n]$ such that for every $i \in [n]$, we have $(v_{\pi(i)}, v_{\pi(i+1)}) \in E$ (with $n+1$ identified with 1).

- A particular graph $G$ may have many HCs, or in some cases (more likely for a sparse graph) no HC.
- Similar definition holds for a directed graph (we require $(v_{\pi(i)}, v_{\pi(i+1)})$ for every $i \in [n]$).
- The problem of deciding whether a given graph contains a HC is NP-complete (also NP-complete for directed graphs). So we don’t expect there is a deterministic polynomial-time algorithm to find a HC in an arbitrary graph (or to decide whether there is one).

Random Graphs - Erdős-Rényi model $G_{n,p}$

- $n$ vertices.
- Some fixed probability $p$.
- For every $i \in [n]$, every $j \in [n] \setminus \{i\}$
  - We flip a coin with biased probability $p$, add $(i, j)$ to $E$ if the flip is successful, don’t add it otherwise.
- All the $(i, j)$ trials are identical (probability $p$) and independently distributed.
- $\mathbb{E}_{n, p}[|E|] = \sum_{i,j \in [n], i \neq j} \text{Pr}((i, j) \in E) = \frac{n(n-1)}{2}p$.
- Expected degree of any vertex $i$ is $(n-1)p$.
- Can use deferred decisions to analyse algorithms/structures on $G_{n,p}$.

Examples of Hamiltonian Circuits (or not)

Intuitively HCs are more likely in denser graphs

RC (2016/17) – Lectures 11 and 12 – slide 3

RC (2016/17) – Lectures 11 and 12 – slide 4
Hamilton cycles in Erdős-Rényi graphs

Theorem (Komlós and Szemerédi (1983))
Suppose we generate $G$ according to $G_{n,p}$. Then the existence of a HC in $G$ is characterised by the value of $p$ in relation to $n$:

$$\Pr_{n,p}[G \text{ has a HC}] \rightarrow \begin{cases} 
0 & \text{if } pn - \ln(n) - \ln\ln(n) \rightarrow -\infty \\
1 & \text{if } pn - \ln(n) - \ln\ln(n) \rightarrow +\infty \\
e^{-e^{-c}} & \text{if } pn - \ln(n) - \ln\ln(n) \rightarrow c 
\end{cases}$$

(in the final case, $c$ being any constant)

- These are all with high probability results, holding with probability $1 - o(1)$ (tending to 1 as $n \to \infty$).
- This kind of result is described as a sharp threshold.
- There is a polynomial-time algorithm to find the HC in the middle case (Bollobás, Fenner and Frieze, 1985).
- We will not prove Komlós & Szemerédi’s result, we will show how to find a HC for $p \geq \frac{40\ln(n)}{n}$ (easier).

RC (2016/17) – Lectures 11 and 12 – slide 5

Algorithm maintains a current path $P$ of the form $v_1, \ldots, v_k$, the current head ($hd$) is $v_k$.

The algorithm chooses the “next” extension edge from $\text{UnUsed}(v_k)$ (adjacent to the head), hoping this will add a new vertex to the path.

Given a graph $G = (V,E)$ generated according to $G_{n,p}$.

Algorithm maintains a current path $P$ of the form $v_1, \ldots, v_k$, the current head ($hd$) is $v_k$.

Intuitively, the algorithm tries to randomly choose an extension edge (adjacent to the head) which adds a new vertex to the path (if not, we then “Rotate”).

Three operations are used to “grow” a path (from a starting vertex):

- “Reverse”
- “Rotate”
- “Extend”

$\text{UnUsed}(v)$ contains theAdjacent edges to $v$ which have not been used to extend from $v$ (originally all adjacent edges).

Assume $\text{UnUsed}(v)$ is randomly shuffled for each $v$.

RC (2016/17) – Lectures 11 and 12 – slide 6

- $\Pr_{n,p}[G \text{ has a HC}] \rightarrow 0$ if $pn - \ln(n) - \ln\ln(n) \rightarrow -\infty$
- $\Pr_{n,p}[G \text{ has a HC}] \rightarrow 1$ if $pn - \ln(n) - \ln\ln(n) \rightarrow +\infty$
- $\Pr_{n,p}[G \text{ has a HC}] \rightarrow e^{-e^{-c}}$ if $pn - \ln(n) - \ln\ln(n) \rightarrow c$

Ideal case. Note that $(v_k, y)$ is a uniform random choice from $\text{Adj}(v_k)$, as the edges were shuffled at the start.

RC (2016/17) – Lectures 11 and 12 – slide 7

Sometimes the randomly chosen extension “loops back” onto $P$ (less ideal). We don’t iterate through other $v_k$ neighbours, we “Rotate”.

RC (2016/17) – Lectures 11 and 12 – slide 8
“Reverse, Extend or Rotate” (Algorithm 5.2)

Algorithm REVERSE EXTEND ROTATE (G = (V, E))
1. for v ∈ V do
2. Used(v) ← {}, UnUsed(v) ← [(v, u) : u ∈ Adj_G(v)].
3. Initialise P with a uniform random vertex, initialise hd also.
   // Throughout P is some v₁ ... vₖ (distinct vertices), hd is v_k //
4. while (P is not a HC and UnUsed(hd) ≠ {}) do
5.   With prob. 1/n, “Reverse” P (and reset hd ← v₁)
6.   With prob. (Used(v₁)/n), choose (vₖ, v₁) ∈ adj Used(hd),
    and “Extend” (and reset hd ← v₁vₖ)
7.   With prob. 1 − (1 + Used(v₁)/n), take the first edge (vₖ, y)
    from UnUsed(hd), and “Extend” or “Rotate” (depends on y).
    Move (vₖ, y) from UnUsed(vₖ) to Used(vₖ).
    Update hd to either y (Extend) or vₖ (Rotate).
8. Check whether P is a HC and return P or “no”.

RC (2016/17) – Lectures 11 and 12 – slide 9

Analysing Algorithm 5.2

▶ Overall, the key operation for building the HC is “Extend”.
  “Rotate” and “Reverse” are helper operations which help the
  analysis go through (well, “Rotate” also helps us get unstuck).
▶ Asking quite a lot to get a full HC on a “run” where we
  more-or-less just add random edges to extend P. So we’ll need
  to run the loop for a super-linear number of steps (Ω(n ln(n)),
  see Theorem 5.16, Corollary 5.17).
▶ Assume for Lemma 5.15, Thm 5.16 that UnUsed(v) is randomly
  generated by adding every possible (u, v) with probability q,
  in random order (can use “deferred decisions” in analysis).
▶ All the UnUsed(·) sets are assumed to be independent for
  proving Lemma 5.15 and Theorem 5.16 (not strictly true,
  fixed in Cor 5.17). Means to an end . . .

RC (2016/17) – Lectures 11 and 12 – slide 10

Lemma 5.15

Supposed we run Algorithm REVERSE EXTEND ROTATE on G = (V, E)
with the UnUsed(v) sets generated independently with probability q
for each possible neighbour, and random orders. Let Vᵢ be the “hd”
vertex after t steps. Then, as long as UnUsed(Vᵢ) ≠ {}, for any u ∈ V,

Pr[Vᵢ₊₁ = u | Vᵢ = uᵢ, . . . , V₀ = u₀] = 1/n.

Proof.
Identical to book.
Easy for v₁, and for any of the P-vertices vᵢ₊₁ such that
vᵢ ∈ Used(hd).
For other u vertices (on P or otherwise), we use the principle of
delayed decisions, plus the assumptions on slide 10.

RC (2016/17) – Lectures 11 and 12 – slide 11

Theorem 5.16

Supposed we run Algorithm REVERSE EXTEND ROTATE on G with the
UnUsed(v) sets generated independently with probability q ≥ 20 ln(n)
for each possible neighbour, and random orderings. Then the
algorithm finds a HC after O(n ln(n)) iterations of the loop at 4.,
with probability 1 − O(n⁻¹).

Proof Failure after 3n ln(n) iterations means that either

- E₁: We did 3n ln(n) iterations without constructing a HC,
  with all UnUsed(hd) sets staying non-empty, or,
- E₂: At least one of the UnUsed(hd) lists became empty
during the 3n ln(n) iterations.

To bound Pr[E₁], we want the prob. of not finding a HC, when at each
step, the next “hd” is uniform from V (guaranteed by Lemma 5.15).
This is the “coupon collector” problem (must hit all n vertices).

RC (2016/17) – Lectures 11 and 12 – slide 12
Theorem 5.16 cont’d.

Proof cont’d. For event $E_1$, probability that any particular $v$ does not become “hd” at some stage over $2n\ln(n)$ iterations, is at most

$$
\left(1 - \frac{1}{n}\right)^{2n\ln(n)} < e^{-2\ln(n)} = \frac{1}{n^2}.
$$

Probability that some $v$ fails to become hd in this window, is at most $\frac{1}{n}$ by the Union Bound.

Need to complete the HC with a closing edge to $v_1$, over remaining $\ln(n)n$ steps. Probability of failure is at most $(1 - \frac{1}{n} \ln(n) < \frac{1}{n}$.

So $Pr[E_1] \leq \frac{2}{n}$.

For event $E_2$, we partition into:

- $E_{2a}$: Some $v$ had at least $9\ln(n)$ edges removed from $UnUsed(v)$ during the $3n\ln(n)$ steps. or;
- $E_{2b}$: Some $v$ originally had $\leq 10\ln(n)$ edges in $UnUsed(v)$.

RC (2016/17) – Lectures 11 and 12 – slide 13

Corollary 5.17

Proof of Theorem 5.16 assumed that the $UnUsed(v)$ sets were all generated independently of each other when they are randomly populated. In the “real world” of $\mathbb{S}_{n,p}$, of course $(u, v)$ would add an entry into two “UnUsed” sets.

Our analysis (essentially) assumes that either $(u, v) \in UnUsed(u)$ or $(u, v) \in UnUsed(v)$ is ok to have the edge in the HC.

See Corollary 5.17 for how to define $p$ so that we can populate the $UnUsed$ lists randomly and independently, with $q \geq 20\ln(n)/n$.

RC (2016/17) – Lectures 11 and 12 – slide 16

Reading and Doing

Next topics coming up is the Probabilistic method and Derandomization, could read some of Chapter 6 to prepare.

Exercises:

- Exercises 5.3 and 5.7 are good reminders of the basic “balls in bins” ideas.
- Analysis at top of slide 13 is essentially the “coupon collector” problem, done for specific $v$, then Union Bound. Compare this to what we got with $2n\ln(n)$ on slide 16 of lecture 6. Surprised?
- Read Corollary 5.17 and understand the details.
- Exploratory exercise on “marking the binary tree” (section 5.8).