Randomized algorithmics

Miklós Krész
University of Szeged
and
InnoRenew CoE
Background

- Research in graph theory, discrete optimization and scheduling
- Industrially motivated projects and projects in industrial cooperation
- Decision support modules integrated into business information systems
- Working on real life data
- Contact: miklos.kresz@domlab.hu
Szeged Informatics is 60 years old

- 1957: applied mathematics program for 3 students
- 1958: the Kalmár logic machine
Szeged

University of Ljubljana
May 16, 2019.
Szeged – The city of sunshine

- The sunniest city of Hungary
- Vibrant cultural life

University of Ljubljana
May 16, 2019.
• Szeged, the university town

- The 4\textsuperscript{th} most populous city in Hungary (170 000 people)
- 2\textsuperscript{nd} biggest university city in Hungary

University of Ljubljana
May 16, 2019.
InnoRenew CoE
Renewable Materials and Healthy Environments
Research and Innovation Centre of Excellence


University of Ljubljana
May 16, 2019.
About the InnoRenew CoE

Our Focus:
• Research into renewable materials and their uses
• Emphasis on bringing other disciplines into the fold
  – Human health, ICT, Cultural Heritage, Business Management, Engineering
• Research topics in ICT
  – Building informatics, Industrial optimization, Data science, Sensor networks and IoT
The class $\mathbf{P}$

**Definition**

A language $L \subseteq \Gamma^*$ is in $\mathbf{P}$ if there exists a DTM $M$ that halts on every input in polynomial time such that for every $x \in \Gamma^*$,

$$x \in L \iff M(x) = 1$$

where $M(x) = 1(0)$ if $M$ accepts (rejects) $x$ in poly time in $|x|$. 
The class \textbf{NP}

Here is an alternative definition of \textbf{NP}.

\textbf{Definition}

A language $L \subseteq \Gamma^*$ is in \textbf{NP} if there exists a polynomial $p : \mathbb{N} \rightarrow \mathbb{N}$ and a polynomial time DTM $M$ such that for every $x \in \Gamma^*$,

$$x \in L \iff \exists u \in \Gamma^{p(|x|)} \text{ s.t. } M(x, u) = 1$$

$u$ is called a \textbf{certificate} for $x$ when $x \in L$ and $u \in \Gamma^{p(|x|)}$ satisfy $M(x, u) = 1$. 
An (simplified) industrial example

Given a decision support system for public transit with the module of bus maintenance scheduling. For this module the inputs are

- The weekly schedule of each bus: B1, …, Bn
- The weekly maintenance time periods (1 hour each e.g.): T1, …, Tm

It is a bipartite matching problem such that Bi and Tj are connected if the maintenance in the Tj time period is possible in the schedule of Bi.

If there is a solution:
- „legality” of the solution can be easily checked

If there is no solution:
- Transport engineers need a certification….

Note: The example is realistic, but the min-cost solution is needed, by applying weighted matching and some real-world extra constraints make the model much more involved.
Solutions for (NP-)hard problems

What is a „good” solution?
What is „efficient”?

It depends on the application!
e.g. call routing, bus scheduling

Algorithm design techniques:
• Exact methods for problems with restrictions
• Approximation algorithms
• Randomized algorithms
• Metaheuristics

University of Ljubljana
May 16, 2019.
Randomized algorithms
Randomness in algorithms: a tour from Monte-Carlo to Las Vegas

How do we define the “Monte Carlo Method”? 

- Literally any algorithm, which use random numbers (or takes random decision)
- Probabilistic Monte Carlo Method – the random numbers simulate directly the physical phenomena we would like to observe (direct simulation by MC)
  - nuclear physics
  - random fluctuations in the telephone traffic
  - flood control and dam construction
  - bottlenecks and queueing systems in industrial production processes
  - study of epidemics
- Deterministic Monte Carlo Method – in problems we can formulate in theoretical language, but cannot solve by theoretical means (MC algorithms)
Random algorithms you learned earlier: QuickSort

Divide-and-Conquer sorting algorithm

- choose a pivot element
- partition by pivot element (smaller; equal; bigger)
- recursion on the partitions
What is overall Time Complexity in Worst Case?
In worst case, each partition divides array such that one side has n/4 elements and other side has 3n/4 elements. The worst case height of recursion tree is $\log_{3/4} n$ which is $O(\log n)$.
The idea of (Monte-Carlo) randomization

*Randomization*: fundamental algorithmic design pattern
(others: divide and conquer, dynamic programming, etc.)

*Principle of randomized algorithms*: random choice, random decision etc. (coin flipping) among the commands

*Outputs of randomized algorithms*: the answer is with „high probability” (but „high probability” is determined by an appropriate bound) („sock” ex.)

Is „high probability” satisfactory? → Repeat the process many times

The probability of wrong answers is decreased to a „technical level” (software, hardware faults etc.)

Sequence of real random bits? → Pseudo-random bits
(good behaviour in practice)
Problem:
Given an integer coefficient polynom $f(x_1,\ldots,x_n)$ by substitution with $\deg f \leq d$.

Is $f$ identically $0$?

Given by substitution? $\rightarrow$ Evaluation: assigning values to the variables + calculation

? $\leftarrow$ The polynom is a „black box”

Ex: $f(x_1,\ldots,x_{2n}) = (x_1+x_2) \cdot (x_3+x_4) \cdots (x_{2n-1}+x_{2n})$

The complete form of $f$ has exponential size ($2^n$ terms), but evaluation takes $n$ additions and $n$ multiplications only.
The „witness” theorem

Witness (for polynom $f(x_1,...,x_n)$ not being identically 0):

an assignment $\alpha = (\alpha_1, ..., \alpha_n)$ to $(x_1,...,x_n)$ such that $f(\alpha) \neq 0$.

Meaning of the witness theorem:
If $f$ is not identically 0 $\Rightarrow$ a randomly selected input will be a witness with high probability.

Theorem (Schwarz lemma).
If $f$ is not identically 0 with $\deg f \leq d$, and $\alpha_1, ..., \alpha_n$ are uniformly distributed pairwise independent elements of the set $\{1, ..., N\} \Rightarrow \text{Prob}(f(\alpha)=0) \leq d/N$.

Intuitively:
Eg. $n=2 \Rightarrow f(x_1,x_2) \Rightarrow$ if $f$ is not identically 0, then $\{(\beta_1, \beta_2) \in \mathbb{R}^2, f(\beta_1, \beta_2)=0\}$ is a curve on the plane. $\Rightarrow$ The curve avoids almost all points of the plane $\Rightarrow$ a randomly chosen point will be a witness with high probability.

Corollary. If $\alpha=(\alpha_1, ..., \alpha_n)$ is a randomly composed vector from the elements of the set $\{1,2,...,2d\}$ and $f$ is not identically 0 with $\deg f \leq d \Rightarrow \text{Prob}(f(\alpha )\neq 0) \geq 1/2$.

Proof. Applying Schwarz lemma with $N=2d \Rightarrow \text{Prob}(f(\alpha)=0) \leq d/2d=1/2 \Rightarrow \text{Prob}(f(\alpha ) \neq 0) \geq 1/2$. 

University of Ljubljana
May 16, 2019.
The randomized method

**Algorithm.**
1. Choose a constant $t$ and let $k=1$.
2. Consider a vector $\alpha=(\alpha_1,\ldots,\alpha_n)$ randomly chosen from $\{1,\ldots,2d\}$ and evaluate $f(\alpha)$ by substitution. If $f(\alpha) \neq 0$, then STOP with the answer „$f$ is not identically 0”.
3. If $k<t$, then let $k=k+1$ and continue with step 2, otherwise STOP with the answer „$f \equiv 0$” with probability $\geq 1 - \frac{1}{2^t}$.

**Theorem.** The above algorithm gives correct outputs for all polynomials $f(x_1,\ldots,x_n)$ with $\deg f \leq d$.

**Proof.** The answer „$f$ is not identically 0” is trivially correct. Repeating the corollary of Schwarz lemma $t$ times independently $\Rightarrow$ if $f$ was not identically 0, then the algorithm would terminate at step 3 with probability $\leq (1/2)^t \Rightarrow$ the probability that the answer „$f \equiv 0$” is correct $\geq 1 - \frac{1}{2^t}$.

Is the prob. $1 - \frac{1}{2^t}$ satisfactory? $\Rightarrow$ Reliable computer: fault once in 1000 years $\Rightarrow$

The fault occurs in the next 0.001 second: $\geq (1/2)^{50} \Rightarrow t \geq 50$ is satisfactory
**Primality test**

**Problem:**
Given a large odd integer $m$. Decide whether $m$ is a prime.

**Application:**
Public-key cryptography, coding theory etc. Eg. RSA cryptosystem.

**RSA** (Rivest-Shamir-Adleman, 1979) cryptosystem: It is based on two current realities (which may change in the future).

1. It is easy to generate large integers (several hundred to a few thousand bits long) which are prime with high probability - and therefore can be used as primes.

2. It is hard to find the prime factors of a large number. The amount of resources this factorization takes is expected to be totally unreasonable given the possible payoff.
The Fermat test

Fermat’s Little Theorem: If $a \neq 0 \in \mathbb{Z}_p$ for some prime $p$, then $a^{p-1} \equiv 1 \pmod{p}$.

Fermat Test.

Input: An odd integer $m > 2$ with binary representation.
(Using binary representation, $a^{p-1} \equiv 1 \pmod{p}$ can be calculated polynomially with respect to $\log_2 m$)

Algorithm:
1. Choose randomly an integer $a$ from $[1, m)$.
2. If $a^{m-1} \equiv 1 \pmod{m}$, then $m$ is „probably prime”, otherwise $m$ is „composite”.

The „composite” answer is correct.

What is the bound of the probability for the „prime” answer?

Problem: pseudo-primes. An integer $m$ is pseudo-prime, if it has no Fermat-witness (integer $a$ from $[1, m)$ with $a^{m-1} \equiv 1 \pmod{m}$) E.g. 561 is a pseudo-prime

University of Ljubljana
May 16, 2019.
Miller-Rabin test

**Definition:** Let \( m \) be an odd integer and consider \( m-1 \) in the form: \( m-1 = 2k \cdot n \), where \( n \) is odd. The integer \( 1 \leq a \leq m \) is a Miller-Rabin witness (for the compositeness of \( m \)), if none of the following numbers can be divided by \( m \):

\[
a^n - 1, a^n + 1, a^{2n} + 1, \ldots, a^{2^{k-1}n} + 1
\]

**Theorem.** If \( m \) is prime, then there does not exist a Miller-Rabin witness for \( m \).

**Theorem.** If \( m \) is composite, then at least half of the integers of \([1, \ldots m)\) are Miller-Rabin witnesses.

**Corollary.** For any random integer \( a \) from \([1, \ldots m)\):

- If \( a \) is not a Miller Rabin witness \( \Rightarrow m \) is composite.
- Otherwise \( m \) is prime with probability \( \geq 1/2 \).
The algorithm

**Miller-Rabin Test.**

**Input:** An odd integer $m > 2$ with binary representation.

**Algorithm:**

1. Decompose $m–1$ in the form: $m–1 = 2^k \cdot n$, where $n$ is odd.
2. Choose randomly an integer $a$ from $[1,m)$
3. Determine whether $a$ is a Miller-Rabin witness.
4. If $a$ is not a Miller-Rabin witness, then $m$ is „probably prime”, otherwise $m$ is „composite”.

Repeating the test $t$ times without Miller-Rabin witness,
the probability that a composite $m$ is identified as prime is $\leq (1/2)^t$. 

University of Ljubljana
May 16, 2019.
Finding large primes

**Problem.** Given an integer $n$. Find a prime with length $n$ in binary representation.

**Method.** Choose a random integer $m$ in $[2^{n-1}, 2^n-1]$. Determine by Miller-Rabin test if $m$ is prime with the probability $2^{-60}$. It takes at most 60 tests. If $m$ is not prime, then choose another $m$.

**How many tests are needed?** After $O(n)$ tests a prime is found with probability approximately 1.
Simulation methods
A „gambling” example

Calculating $\pi$ on a dartboard

- Take a square frame of size $2 \times 2$, and place a circle dartboard $r = 1$ on it.
- What is the probability of hitting the dartboard of those hits inside the frame?
  - We consider only the upper-right quarter for simplicity of calculation.
  - 140 hits of 180: $140/180 = 0.778$.
- The probability is the ratio of the areas:
  - $A_{\text{circle}} / A_{\text{square}} = r^2 \pi / s^2 = 1^2 \pi / 2^2 = \pi / 4 = 0.7853981$. 
Calculation by simulation

- We generate numbers for $x, y$ coordinates in the upper-right corner
- $\rightarrow$ two random numbers in the range $[0, 1]$
- Is it a hit to the dartboard?
- Calculate the distance to the origin: $\sqrt{x^2 + y^2}$
- If the distance smaller than the radius $r = 1$, then it is a hit!
- (instead of square root calculation we raise both sides to the square and use: $x^2 + y^2 < 1$)
Suppose we want to calculate the integral of a smooth function $f$ on the interval $[a, b]$ on the real axis:

$$ I = \int_{a}^{b} f(x) \, dx $$

In numerical methods we choose points in the desired interval and interpolate the function using function values at these points. We use $N$ equidistant points starting from $a$:

$$ x_n = a + n\frac{(b-a)}{N}.$$
Using the rectangular formula

\[ \int_{a}^{b} f(x) \, dx \approx \frac{(b - a)}{N} \sum_{n=0}^{N-1} f(x_n) \]
Using the trapezoidal formula

\[ \int_{a}^{b} f(x) \, dx \approx \frac{(b - a)}{N} \left( \frac{1}{2} \sum_{n=0}^{N-1} (f(x_{n+1}) + f(x_n)) \right) \]
Monte Carlo Integration

The techniques described called the Quadrature Formulas, where we choose some $w_n$ weights for the integration:

$$\int_a^b f(x) \, dx \approx \frac{(b - a)}{N} \sum_{n=0}^{N-1} w_n f(x_n)$$

The Monte Carlo Integration is a variation of this method, where we choose all the weights $w_n = 1$, and we choose the $x_n$ points randomly! The benefits are:

- In higher dimension the exact numerical methods need so many points that we cannot calculate with them
- In higher dimension the exact numerical methods tend to be less and less accurate, while the error of the Monte Carlo Integration is independent of dimension: $O(\sqrt{1/N})$
- The Monte Carlo method is easy to implement, and...
Categorization

We can categorize the deterministic Monte Carlo Method by the nature of its error.

- **Two sided error**
  - typical for engineering simulations (the MC integration is an example)
  - we have a ± error
  - the magnitude of the error controlled by the number of sampling points
  - we can stop at any time

- **One sided error**
  - the primality tests are good examples
  - we “ask” something, and get a probability answer with one sided error
  - if the answer is “not prime”, it is 100% certain, if the answer is “prime”, it is probable
  - we can speak about one sided error on true side, or false side

→ easy parallel implementation for both (mostly the first)
Categorization (cont.)

- Zero sided error
- $\rightarrow$ the algorithm runs with no error at all
- QuickSort is a good example – we always get a sorted sequence
- Definition: $A$ is a Las Vegas algorithm for a problem class $\Pi$, if and only if
  - if for a given problem instance $\pi \in \Pi$, algorithm $A$ terminates returning solution $s$, $s$ is guaranteed to be a correct solution of $\pi$
  - for any given instance $\pi \in \Pi$, the run-time of $A$ applied to $\pi$ is a random variable
- (we can speak of certainly terminating algorithms as well)

These algorithms called the Las Vegas algorithms
Random walks in graphs

PageRank: Google (1998) implemented

Importance order of $N$ webpages

The number of visitors? No! (No audited)

Links: if we place a link on our website, then the linked page is important for us

Idea:

a, if „numerous” links to a page → „important” page
b, important page links a page→ also „important” apge

$$A_{N \times N} = \begin{cases} \frac{1}{n} & \text{if } i \text{ links to } j \\ \frac{n}{n} & \text{if there are } n \text{ links from page } i \\ 0 & \text{otherwise} \end{cases}$$

Row-stochastic matrix: rowsum is 1 in each row & $A_{ij} \geq 0$
Theorem: If $A$ is a row-stochastic matrix of $N \times N$ & $j = (\frac{1}{N}, \ldots, \frac{1}{N})$

$$\downarrow$$

$$p = \lim_{m \to \infty} j A^m$$ exists &

$$pA = p$$

Definition: Vector $p = (p_1, \ldots, p_N) \in R^N_+$ is the rank-vector of the pages

(thus the rank of page $i$ is $p_i$)

Algorithm:

1. Construction of matrix $A$ from the topology of web pages

2. Initializing: $p = (\frac{1}{N}, \ldots, \frac{1}{N})$

3. $p_{i+1} := p_i \ast A$

4. If stop criterium is fulfilled $\rightarrow$ STOP

   Otherwise goto 3.

Stop criterium: If the rank vector $p$ changes below a bound.

$$\downarrow$$

The order of the components is important, not the value. $\rightarrow$ OK, if the order is stable.

University of Ljubljana
May 16, 2019.
Intuitively:

„Stochastic web surfer”:
Starting point is random, the next one is chosen randomly through links

After infinite no of steps arriving to page $i$ with probability $p_i$

Advantage: fast (mtx products efficiently calculated) and easy to implement

$N * j * A^m$
Example:

\[ \xrightarrow{X} \xrightarrow{Z} \xrightarrow{Y} \]

Intuitively:

„Stochastic web surfer”:
Starting point is random, the next one is chosen randomly through links

After infinite no of steps arriving to page \( i \) with probability \( p_i \)

Advantage: fast (mtx products efficiently calculated) and easy to implement

\[ N \times j \times A^m \]
Algorithm problems:

1. „Dead end” problem:
   Dead end: page from which there is no link, but other page likes it
   ↓
   This row of $A$ consists of 0 only $\rightarrow A$ is not row-stochastic $\rightarrow$
   algorithm works incorrectly, the importance of the pages „slipes out”
   from the system

Example:

$$
A = \begin{pmatrix}
\frac{1}{2} & \frac{1}{2} \\
\frac{1}{2} & \frac{1}{2}
\end{pmatrix}
\rightarrow A^2 = \begin{pmatrix}
\frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{4}
\end{pmatrix} = \frac{1}{2} A \rightarrow \cdots \rightarrow A^m = \frac{1}{2^{m-1}} \times A \rightarrow
$$
Problem of spider traps:

System of pages, where each links direct inside the system.

\[ \downarrow \]

Collect „the importance.”

\[ \downarrow \]

SPAM, abuse etc. By removing links, everybody can produce this kind of problems.
Example (earlier one modified):

\[
\begin{pmatrix}
\frac{1}{2} & 0 & \frac{1}{2} \\
0 & 1 & 0 \\
\frac{1}{2} & \frac{1}{2} & 0
\end{pmatrix}
\rightarrow \text{a Rank-vector (multiplied by } N) :
\]

\[
N*p_1 = (1,1,1)
\]
\[
N*p_2 = (1,\frac{3}{2},\frac{1}{2})
\]
\[
N*p_3 = (\frac{3}{4},\frac{7}{4},\frac{1}{2})
\]
\[
N*p_4 = (\frac{5}{8},\frac{2}{8},\frac{3}{8})
\]
\[
N*p_5 = (\frac{1}{2},\frac{35}{16},\frac{5}{16})
\]

\[
p = (0,1,0) \text{ (can be proved)}
\]
The „real” Page Rank

Eliminating the „Dead end” and „Spider trap” problems:

„Taxing” the pages

Collecting from each page a certain part $\varepsilon$ of its importance, then this importance is distributed uniformly.

Instead of $A$ we work with the following matrix.

$$B = \varepsilon \cdot U + (1 - \varepsilon) \cdot A$$

$$U = \begin{pmatrix}
\frac{1}{N} & \cdots & \frac{1}{N} \\
\vdots & & \vdots \\
\frac{1}{N} & \cdots & \frac{1}{N}
\end{pmatrix}$$

$B$ is row-stochastic, thus convergence is guaranteed.
Thank you for your attention!