THE UNIVERSITY OF WESTERN AUSTRALIA
Achieving International Excellence

Estimation, Probability Bounds, and Complexity of Algorithms

Cheryl E Praeger

Aachen, July, 2019

Briefly: aim of lecture

- Link: estimation/randomisation
- Two simple examples for estimation and algorithms
    - in Permutation groups
    - in classical matrix groups
- A “going down” algorithm in linear groups

Randomisation - Why?

THE UNIVERSITY OF WESTERN AUSTRALIA
Achieving International Excellence

Some potted history

- Charles Sims' permutation group algorithms

Base of permutation group $G \leq S_n$

- A sequence of points $(i_1, \ldots, i_r)$ such that $G_{i_1,\ldots,i_r} = 1$
- Distinct $g, g' \in G$ correspond to distinct base images
  - $(i_1, \ldots, i_r)g$ and $(i_1, \ldots, i_r)g'$
  - Only need to know action on r points, not all n points
- Example $G = D_{2n} = \langle a = (12 \ldots n), b = (2n)(3, n-1) \ldots \rangle$,
  - Base $B = (1,2)$ so each $g \in G$ determined by $(1g, 2g)$
- Small bases give compact [space/time saving] in computations

Sims' ingenious methods compute using base images

Still – Why randomisation?

THE UNIVERSITY OF WESTERN AUSTRALIA
Achieving International Excellence

Usefulness [around 1970]

- Sims proved existence of Lyons sporadic simple group by constructing it as a permutation group on 9×10^6 points (smallest possible) on a computer which could not even store and multiply the two generators! He needed to use base images

So what's the problem?

- Sims general purpose perm group algorithms great
- Except when minimum base size too large
- The Giants: S_n and A_n
- Base for S_n – $(1,2,\ldots,n-1)$
- Base for A_n – $(1,2,\ldots,n-2)$

THE UNIVERSITY OF WESTERN AUSTRALIA
Achieving International Excellence

John Cannon and CAYLEY 1970s

- Given $G = \langle X \rangle$ permutation group with gen'g set X
  - If G is primitive and not A_n or S_n then G has a much smaller base and Sims' methods worked brilliantly [for computations then]
  - For A_n or S_n need special methods

- So how to identify the giants A_n and S_n ?
  - Use theory from 1870s
  - Many elements ONLY exist in giants
  - So many that we should find them with high probability by random selection in a giant

THE UNIVERSITY OF WESTERN AUSTRALIA
Achieving International Excellence

Jordan's Theorem circa 1870

- Given transitive permutation group $G \leq S_n$, and a prime p such that $\frac{n}{2} < p < n - 2$
- If some element of G contains a p-cycle then G is A_n or S_n

How useful is this?

How common are Jordan's 'good' elements?

Define: $g \in S_n$ is **'good'** if g contains a p-cycle, for some prime p, $n/2 < p \leq n-3$

Example: $g = (12345)(67) \in S_9$ is 'good': $n = 9, p = 5$

For fixed p: number of elements in S_n containing a p-cycle is

$$\binom{n}{p}(p-1)!(n-p)! = \frac{n!}{p} \quad \text{(and} \quad \tfrac{n!}{2p} \quad \text{in } A_n)$$

Proportion of 'good' elements in A_n or S_n: $\sum_{n/2<p\leq n-3} \frac{1}{p} \geq \frac{c}{\log n}$ for some constant c

So roughly c from every log n elements is "good"
Develop this into a "justifiable algorithm"

Monte Carlo algorithm to recognise S_n, A_n

Input: Transitive $G = \langle x_1, \ldots, x_k \rangle \leq S_n$ and real number ε ($0 < \varepsilon < 1$, error probability bound)

Output: True (hopefully if G is S_n or A_n) or False

Algorithm: Select up to $N = \lceil (\log \varepsilon^{-1})(\log n)/c \rceil$ random elements g from G and test if g is 'good'.

If a 'good' element is found then return True
If no 'good' elements are found then return False

What does this algorithm actually do?: (At least it completes!)

1. If the algorithm returns True then $G = A_n$ or S_n (**guaranteed by Jordan's Theorem**)

2. If the algorithm returns False then this may be incorrect, but only if G does equal A_n or S_n, and we failed to find a 'good' element.

3. **Prob**(do not find good element, **given that $G = A_n$ or S_n**)

$$\leq \left(1 - \frac{c}{\log n}\right)^N < \varepsilon$$

So this is a **Monte Carlo algorithm with error probability less than** ε.

THE UNIVERSITY OF WESTERN AUSTRALIA
Achieving International Excellence

Monte Carlo algorithms

- named after Monte Carlo Casino in Monaco
- where physicist Stanislaw Ulam's uncle used to borrow money to gamble

want the algorithm to complete quickly, allow a small (controlled) probability of error.

THE UNIVERSITY OF WESTERN AUSTRALIA
Achieving International Excellence

Monte Carlo algorithms

- named after Monte Carlo Casino in Monaco
- where physicist Stanislaw Ulam's uncle used to borrow money to gamble

Famous uses:

- Enrico Fermi (1930) the properties of the neutron
- Los Alamos (1950s) for early work on hydrogen bomb

Further Comments on Context

1: assume available approximately independent **random elements** from G (Both theoretical and practical algorithms exist for this.)

2: Monte Carlo algorithms: error probability must be controlled

3: variety of mathematics required for both design and proof

Algebra	to prove correctness of output
Probability estimates	to control error

- This is `essentially' algorithm used in GAP and MAGMA for testing if G is a permutation group giant. Developed by John Cannon.
- Cannon's algorithm relies on generalisations of Jordan's Theorem due to Jordan, Manning, CEP and others. Use a larger family of `good‘ elements.
- Might have seen new paper by Bill Unger on ArXiv

Notice the role of estimation:

lower bound for proportion of “good” elements leads to upper bound on error probability

How good an estimate?

Do we need?
Should we work for?

If estimate is far from true value does it matter?

- Yes and No !
- No: because if there are more good elements than we estimate then we just find them more quickly and algorithm confirms “G is a giant” more quickly
- Yes: because if G is not a giant then we force the algorithm to do needless work in testing too large a number of random elements [it will never find a good one] and so the algorithm runs too slowly!

So the upshot is: it really does matter. We should try to make estimates as good as possible, especially when they are for an algorithmic application.

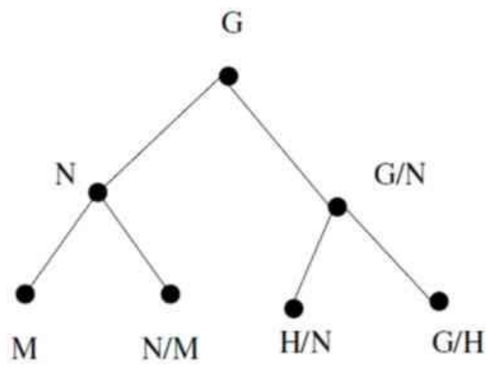
The University of Western Australia
Achieving International Excellence

General group computational framework focuses on simple groups

Few general statements on group computation

'Tree View' underpins new generation of group algorithms:

Focus on finite simple groups:



Some names: O'Brien, Leedham-Green, Seress, Neunhoeffer, . . .

Example from classical groups

$\mathrm{Class}(n, q) = \mathrm{GL}(n, q), \mathrm{Sp}(n, q)$ etc acting on $V = V(n, q)$

Primitive prime divisor (ppd) of $q^e - 1$ a prime r dividing $q^e - 1$ such that $\nexists\, i < e$ with r dividing $q^i - 1$

Ppds interesting because superficially

$$|\mathrm{Class}(n, q)| = q^{\text{some power}} \prod_{\text{various } i} (q^i - 1)$$

ppd-$(n, q; e)$ element $g \in \mathrm{Class}(n, q)$ is an element with order divisible by a ppd of $q^e - 1$;

"good ppd element": $e > n/2$ plus minor additional conditions

1998 Alice Niemeyer and CEP: ppd Classical Recognition Theorem

For an irreducible subgroup G of Class(n, q), if G contains “two different good ppd elements” then essentially G = Class(n, q) with SMALLLIST of exceptions

Deep result – proof relies on simple group classification

Classical recognition algorithm 1998 [NieP]

Input: $G = \langle X_1, \ldots, X_k \rangle \leq \mathrm{Class}(n, q)$

Output: True (and then sure that $G = \mathrm{Class}(n, q)$), or False.

Classical Recognition Algorithm: Niemeyer, CEP, 1998

1. Test MANY random elements of G;
2. If "two good ppd elements" not found return False;
3. If found and test for membership in SMALLLIST positive, return False;
4. Else report True

But how many is MANY?

Is it really a Monte Carlo algorithm?

- If it returns True then G really is $\mathrm{Class}(n, q)$ (by theorem)
- If it returns False this may be incorrect

  (namely if $G = \mathrm{Class}(n, q)$ and we fail to find good ppds).

If we knew the proportion of "good ppd pairs" in $\mathrm{Class}(n, q)$ then we could estimate how many random elements to test – Monte Carlo Algorithm

Basic problem: Estimate the proportion of good ppd elements in $\mathrm{Class}(n, q)$.

First the answer:

For $G = \mathrm{Class}(n, q)$ and $e > n/2$ let $\mathrm{PPD}(G, e)$ be the proportion of ppd-$(n, q; e)$ elements in G

Adding over all such e let $\mathrm{PPD}(G)$ be proportion of ppd elements in G

ppd Estimation Theorem: Niemeyer, CEP, 1998

Let $e > \frac{n}{2}$ such that $q^e - 1$ divides $|G|$. Then

(a) $\frac{1}{e+1} \leq \mathrm{PPD}(G, e) \leq \frac{1}{e}$.

(b) $\log 2 - \frac{2}{n} \leq \mathrm{PPD}(G) \leq \log 2 + \frac{2}{n}$

[or half this for some types of classical groups]

The Estimation result uses geometry and group theory (not the FSGC)

- Need only a constant number $c = c(\varepsilon)$ random selections to find a ppd-pair with probability at least $1 - \varepsilon$
- Case G=GL(n,q) – others similar -- For fixed e first find PPD(G,e) same as for G=GL(e,q)
- Show this is (1/e) x (proportion of such elements in cyclic group of order q^e-1)

Fast Forward:

- 2009 Leedham-Green & O'Brien & Lubeck & Dietrich: Constructive recognition of $G = Cl(d,q)$ for q odd.
  - Involves construction of balanced involution centralisers: Colva will speak about this.
- 2011 Akos Seress & Max Neunhoeffer: general q
  - REPACEMENT for balanced involutions: must be easy to find; have good generation properties.
  - A major facet of constructive recognition algorithms: find small classical subgroups – such as SL(2,q) with (d-2)-dim fixed point space.

Fast Forward:

- **Crucial Ideas belong to Akos:** Akos proposed:
  - use "good-ish elements" t in Cl(d,q) - like "tadpoles"
    - Large fixed point space F
    - Irreducible on t-invariant complement U with dim U = n
    - Wanted also order of $t|_U$ divisible by ppd of $q^n - 1$
- Akos believed: with high probability, two random, conjugate good-ish elements t, t' generate $\langle t, t' \rangle$ a Classical group of dimension 2n (and fixed point space of dimension d-2n)

THE UNIVERSITY OF WESTERN AUSTRALIA
Achieving International Excellence

Consequence:

- **So in one step, descend from dimension d to dimension 2n**
- **Akos adamant: we could take n ~ log d**
  1. Must be easy to find; are they?
  2. Must have good generation properties; do they?
- 1 – an estimation problem – I'll discuss this
- 2 – needs FSGC, delicate algorithm development – work still on-going

Consequence:

- 1 – an estimation problem – I'll discuss this

Alice Niemeyer & CEP, published 2014

- Elements in finite classical groups whose powers have large. *Disc. Math. and Theor. Comp. Sci.* **16,** 303-312. arXiv:1405.2385.

- 2 – needs FSGC, delicate algorithm development – work still on-going

CEP & Akos Seress & Sukru Yalcinkaya 2015

- Generation of finite classical groups by pairs of elements with large fixed point spaces, *J. Alg.* **421,** 56-101. arXiv: 1403.2057

The estimation problem

- Random $g \in Cl(d,q)$ with characteristic polynomial c(x).
  - Want c(x) = f(x) h(x) with
    - f irreducible of degree n between log d and 2 log d,
    - **f does not divide h**,
    - so t:= h(g) fixes $V = F \oplus U$ where $F = fix_V(t)$ and $t|_U$ irreducible,
    - and **Akos also wanted $t|_U$ to be a ppd-element**
  - What Akos wanted he got!

The estimation problem

- Random $g \in Cl(d, q)$ with characteristic polynomial c(x).
  - Want c(x) = f(x) h(x) with
    - f irreducible of degree n between log d and 2 log d,
    - all irreducible factors of h have degree coprime to n
    - so a power t of g fixes $V = F \oplus U$ where $F = fix_V(t)$ and $t|_U$ irreducible,
    - and Akos also wanted $t|_U$ to be a ppd-element
  - Alice and I proved: Probability of these conditions holding for a random g is > $\frac{c}{\log d}$

Applications in black box setting

THE UNIVERSITY OF WESTERN AUSTRALIA
Achieving International Excellence

Thank you