Linear-time Algorithms for Proportional Apportionment

David Eppstein

(Joint work with Zhanpeng "Jack" Cheng)

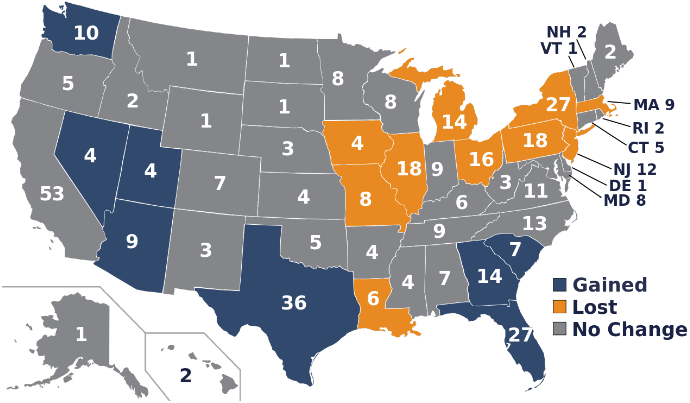
IMBS Colloquium, May 12, 2016

United States congressional apportionment

Every ten years:

The census bureau counts people living in the U.S.

States are given seats in congress, proportional to their population

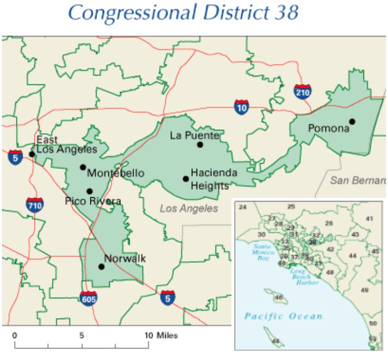


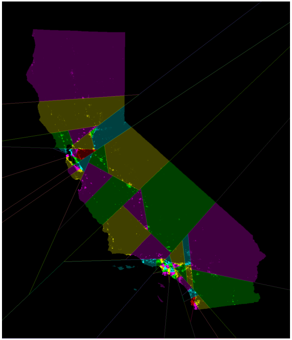
Each state determines its congressional district boundaries

Redistricting

Handled differently at different times in different states

Interesting topics for additional algorithmic research:
how to quantify fairness and automatically draw fair districts?





http://rangevoting.org/SplitLR.html

But this is beyond the scope of today's talk

What does proportional to population mean?

U.S. population / congressional seats:

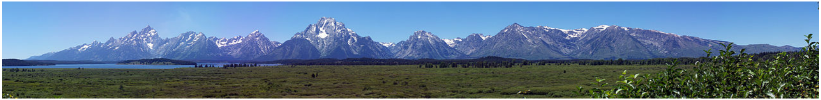
$$\frac{3.19 \times 10^8}{435} \approx 7.33 \times 10^5 \text{ people/seat}$$

California population / congressional seats:

$$\frac{3.88 \times 10^7}{53} \approx 7.32 \times 10^5 \text{ people/seat}$$

Wyoming population / congressional seats:

$$\frac{5.84 \times 10^5}{1} \approx 5.84 \times 10^5 \text{ people/seat}$$



Grand Tetons Panorama by Little Mountain 5 from Wikimedia commons

Solution: Rounding

Since 1913, total # seats is exactly 435

(with one temporary exception for HI+AK)

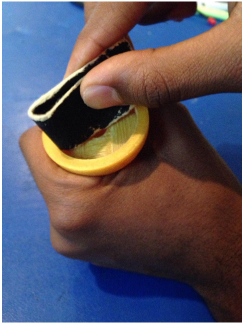
So:

$$\text{seats/state} \approx \frac{435 \times \text{state population}}{\text{US population}}$$

This is not usually an integer!

Rounding to nearest integer could give wrong # seats, shut out small states

Instead we need a rounding rule that always hits the target # seats exactly (and guarantees ≥ 1 seat/state)



Sanding the Nozzle
by Yonatanadane
from Wikimedia commons

Related: Party-list proportional representation

For many countries' elected bodies:

- A fixed number of seats are open for election
- Each party provides a slate of candidates
- Each voter chooses a party or one of its candidates
- Seats are assigned to parties in proportion to their vote count

Key difference: Small parties might not win any seats

ElezioneBrunate by Kaihsu Tai
from Wikimedia commons

Mathematics of approximation by round fractions

This general area is called "Diophantine approximation"
and it has many applications

A famous example: $\pi \approx \dfrac{355}{113}$ [Zu Chongzhi, 5th cent.]

But in mathematics, accuracy is the main goal

In politics, other goals include fairness, inclusiveness, representativity, etc.

Diophantine approximation in music

Piano divides (logarithmic) pitch space into steps of 1/12 octave

Perfect fifth should be $\log_2 \frac{3}{2}$, very accurately approximated as $\frac{7}{12}$

Minor and major thirds are ok but less accurate,

$$\log_2 \frac{6}{5} \approx \frac{3}{12} \text{ and } \log_2 \frac{5}{4} \approx \frac{4}{12}$$

Rhodes Mark II Stage Piano by Tumpatumcla from Wikimedia commons

Mathematical formulation of apportionment

Input: real numbers x_i (populations or votes)
and an integer s (how many seats to assign)

Output: integers a_i with $\sum a_i = s$ and $\frac{x_i}{\sum x_i} \approx \frac{a_i}{s}$
(possibly with extra conditions e.g. $a_i > 0$)

Baling straw, near Southfield Farm by Philip Halling from Wikimedia commons

Two major classes of apportionment methods

Divisor (quota) methods:

Seats = round(population $/D$)

$$D \approx \frac{\text{total population}}{\text{total seats}}$$

Adjust D to make rounded seat count come out correct

Highest averages methods:

Assign seats one at a time, with priority =

$$\frac{\text{population}}{\text{function}(\#\text{ seats so far})}$$

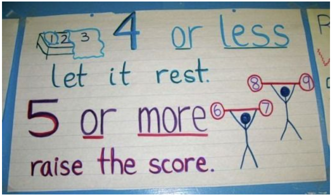
Mathematically equivalent!

Double Road Panorama by Dreamy Pixel from Wikimedia commons

How to round in divisor methods?

D'Hondt–Jefferson:
Round down to an integer, discarding any fractional part

Webster–Sainte-Laguë:
Round to the nearest integer (rounding up for half-integers)

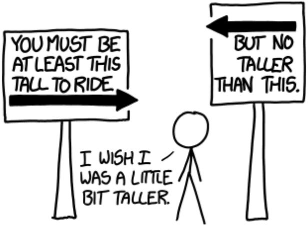


Huntington–Hill:
Round away from geometric mean of the nearest integers
$(0, \sqrt{2}) \Rightarrow 1$; $(\sqrt{2}, \sqrt{6}) \Rightarrow 2$; $(\sqrt{6}, \sqrt{12}) \Rightarrow 3$; ...

Barrage methods (barrier to entry):
Modify rounding rule so a wider interval of numbers rounds to zero

Equivalence of divisors $\Leftrightarrow$ highest averages

- Allocate seats by round(pop $/D$)
  but start with $D = +\infty$ so nobody gets any seats
- Gradually decrease D, allocating one seat at a time
- Each seat is allocated to the state or party
  with the biggest value of population $/f(\#$ seats allocated)
  where $f(n) =$ lower threshold for rounding to $n+1$



https://what-if.xkcd.com/77/

Highest averages formulations of same methods

D'Hondt–Jefferson:

$$\text{priority} = \frac{\text{population}}{s+1}$$

Webster–Sainte-Laguë:

$$\text{priority} = \frac{\text{population}}{2s+1}$$

Huntington–Hill:

$$\text{priority} = \frac{\text{population}}{\sqrt{s(s+1)}}$$

Barrage methods (barrier to entry):

Decrease priority(0)

One more method that doesn't quite fit

Hare–Niemeyer, Vinton, Hamilton, or largest remainder method:

- $D = \frac{\text{total population}}{\text{total seats}}$
- seats = round(population / D)
- If rounding all fractions down would leave v vacant seats, then round up the largest v fractional parts, round down the rest

Why use this class of methods?

Alabama paradox: increasing the number of available seats can decrease an individual state or party's allocation

- From 1852, Congress used largest remainders (replacing Jefferson's method)
- In 1880, increasing total seats from 299 to 300 would have decreased the seats for Alabama from 8 to 7
- Related problems re-occurred in 1900 (involving Maine vs. Virginia)
- In 1910 Congress switched to Webster
- In 1941 switched again to Huntington–Hill

Only divisor methods avoid this issue [Balinsky and Young 1982]

Which particular method to choose?

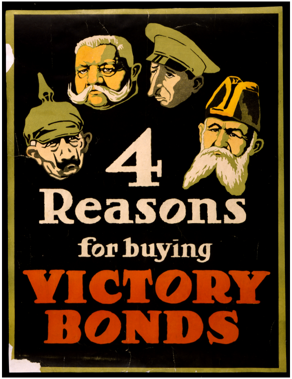
Good criteria:

- By which social criteria it fits best
- By how well it performs in practice
- By how easily it can be understood by participants

Bad criterion:

- By how quickly it can be calculated

My goal: Make them all so quick that nobody would use the bad criterion



History of apportionment complexity research

Naïve methods for both divisor and highest averages formulations

Long known and used, complexity not analyzed

Priority queue data structure for highest averages

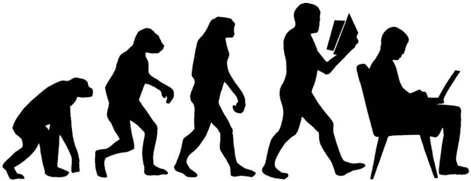
Mentioned in a survey of apportionment by R. B. Campbell (2007)

Linear time selection algorithms

Ito and Inoue (2004, 2006): In Japanese

Cheng and Eppstein (2014)

Simplification by Reitzig and Wild (2015)



Evolution-des-wissens by Phillip Wilke (WMDE) from Wikimedia commons

How fast is fast enough?

For actual vote counting: probably doesn't matter
Any calculation will be dominated by physical vote collection

For use as a subroutine in repeated simulations
(e.g. to test effects of polling errors on vote outcomes):
Faster is always better

2015-09-29 09 34 14 An 80 miles per hour speed limit sign along eastbound Interstate 80 about 31.0 miles west of the Nevada state line in the Bonneville Salt Flats of Tooele County, Utah by Famartin from Wikimedia commons

Comparing algorithm speed to input parameters

Number of steps proportional to population (p): slow

Number of steps proportional to seats (s): intermediate speed

Number of steps proportional to parties or states (n): fast

Matches input size (n vote counts) and and output size (n seat counts) so can't hope to be faster

http://www.gutenberg.org/etext/19994

An intermediate-speed naïve algorithm

- Initialize a priority queue data structure with priorities from the highest averages formulation
- Repeatedly select the state or party with the highest priority, give it a seat, and calculate its new priority

$$\# \text{ steps} = s$$

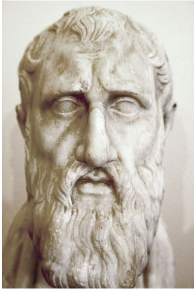
$$\text{Time} = O(s \log s)$$

Americana Scarecrow (516752575) by Steve Evans from Wikimedia Commons

Binary search

Another attempt at a faster naïve algorithm...

- Search for the adjusted value D of the divisor method, starting with a wide interval containing the correct value
- Repeatedly set $D =$ middle of interval, compute round(population$/D$), test if this gives too many or too few seats
- Continue in upper or lower half-interval
- Stop when finding D that gives the correct number of seats



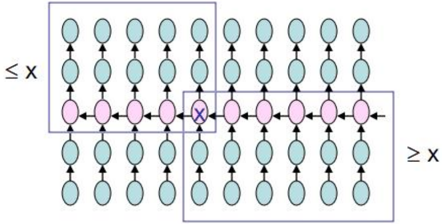
Zeno of Citium - Museo archeologico nazionale di Napoli by Jeremy Weate from Wikimedia Commons

Time: $O(n)$ per bisection
But may repeat infinitely many times! (e.g. for tied priorities)

Selection

Largest remainders method needs to find q largest of n values (fractional parts of population $/D$, with $q =$ remaining seats)

This is a classical and well-studied problem in computer science!



Textbook solution: repeatedly group into 5-tuples, find median of medians recursively, use it to eliminate $\geq 3n/10$ of the values

Other more-practical linear time solutions known (e.g. quickselect)

Highest-averages methods as a selection problem

It's convenient to invert the problem:

- Instead of assigning seat to max population $/f(\#$ seats$)\dots$
- Assign it to min $f(\#$ seats$)/$ population

For each state/party, list $f(i)/$ pop for $i = 0, 1, \dots s - 1$

Then, choose the smallest s of these $n \times s$ values

CA	0.02577	0.05155	0.07732	0.10309	...
TX	0.03704	0.07407	0.11111	0.14815	...
FL	0.05000	0.10000	0.15000	0.20000	...
NY	0.05051	0.10101	0.15152	0.20202	...

But, how to do this quickly, without calculating all $n \times s$ values?

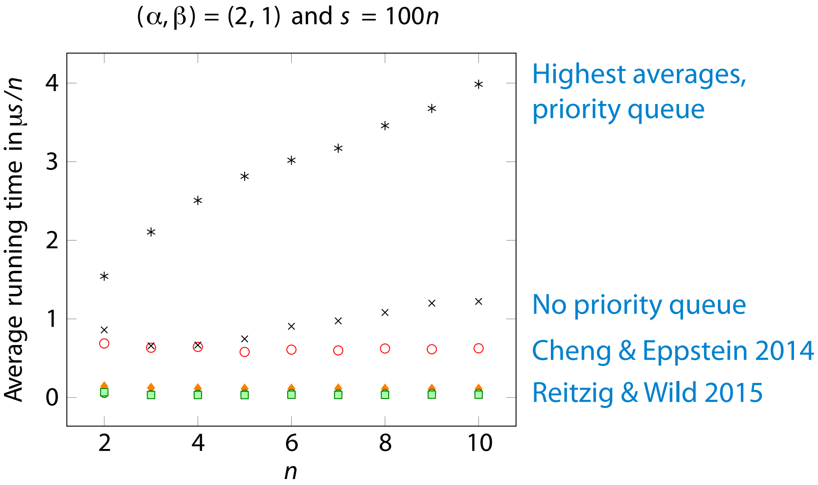
Highest averages in linear time

Idea: we're not sure exactly where the line between allocated and unallocated will be, but we can make an accurate estimate that eliminates most of the work

- $f(\#\text{ seats})$ will be nearly-linear
- Estimate how many $f(i)/\text{pop}$ are below a given threshold x as $\sum f^{-1}(x \cdot \text{pop})$ (also linear)
- Invert estimate to find threshold x with estimate$(x) = s$
- Only $O(n)$ values $f(i)/\text{pop}$ near x need to be checked (the ones with i near $f^{-1}(x \cdot \text{pop})$; other values are definitely above or below optimal threshold)
- Use linear-time selection on them

Korea DMZ by Rishabh Tatiraju from Wikimedia commons

Experimental timing on synthetic data



(Image modified from Figure 1 of Reitzig & Wild 2015)

Conclusions

All standard apportionment methods can be made to run in time linear in the input and output size (number of states or parties)

It is possible to simultaneously achieve fast practical performance and guaranteed avoidance of pathological behavior

Likely there are many other computational problems in social/political science whose complexity remains unexplored

Next natural target:
Automated redistricting and fairness evaluation

References

Russell B. Campbell. The Apportionment Problem. In John G. Michaels and Kenneth H. Rosen, editors, *Applications of Discrete Mathematics, Updated Edition*, chapter 1, pages 2–18. McGraw-Hill Higher Education, New York, NY, USA, 2007.

Zhanpeng Cheng and David Eppstein. Linear-time algorithms for proportional apportionment. In *Proc. 25th Int. Symp. Algorithms and Computation (ISAAC 2014)*, Lecture Notes in Computer Science, pages 581–592. Springer-Verlag, 2014. doi: 10.1007/978-3-319-13075-0_46.

A. Ito and K. Inoue. Linear-Time Algorithms for Apportionment Methods. In *Proceedings of EATCS/LA Workshop on Theoretical Computer Science*, pages 85–91, University of Kyoto, Japan, February 2004.

A. Ito and K. Inoue. On d'Hondt Method of Computing. *IEICE Transactions D*, pages 399–400, February 2006.

Raphael Reitzig and Sebastian Wild. A practical and worst-case efficient algorithm for divisor methods of apportionment. Electronic preprint arxiv:1504.06475, 2015.