A dynamical system that solves hard problems

Veit Elser
Cornell

A Christmas card with a cryptographic twist for charity

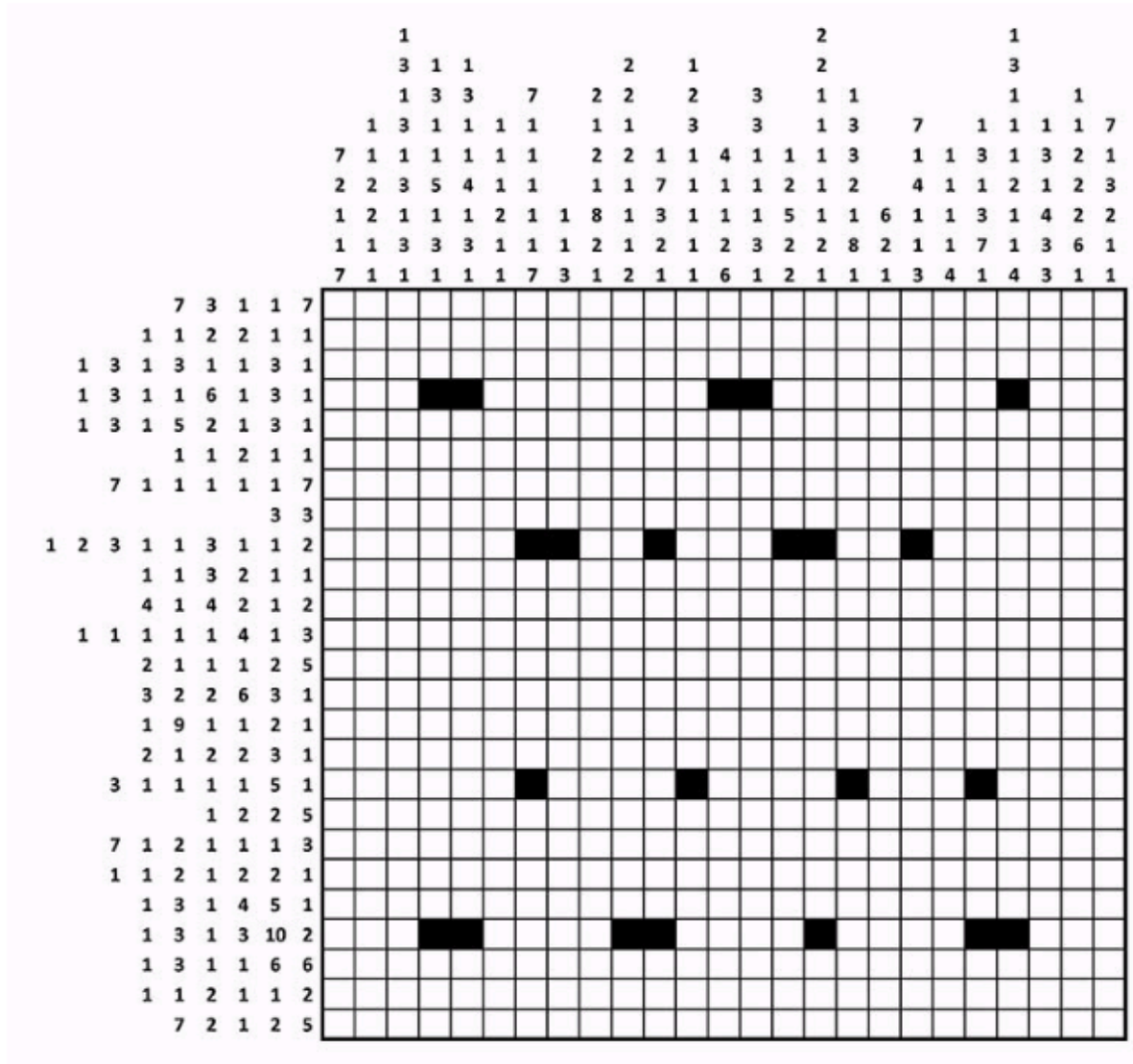
GCHQ Director Robert Hannigan uses his annual Christmas greetings to set a seasonal challenge for charity.

PUBLISHED

07 December 2015

NEWS TYPE

News



(demo)

the general problem ...

$$\text{find } x \in A \cap B$$
$$\text{using projections } P_A,\ P_B$$

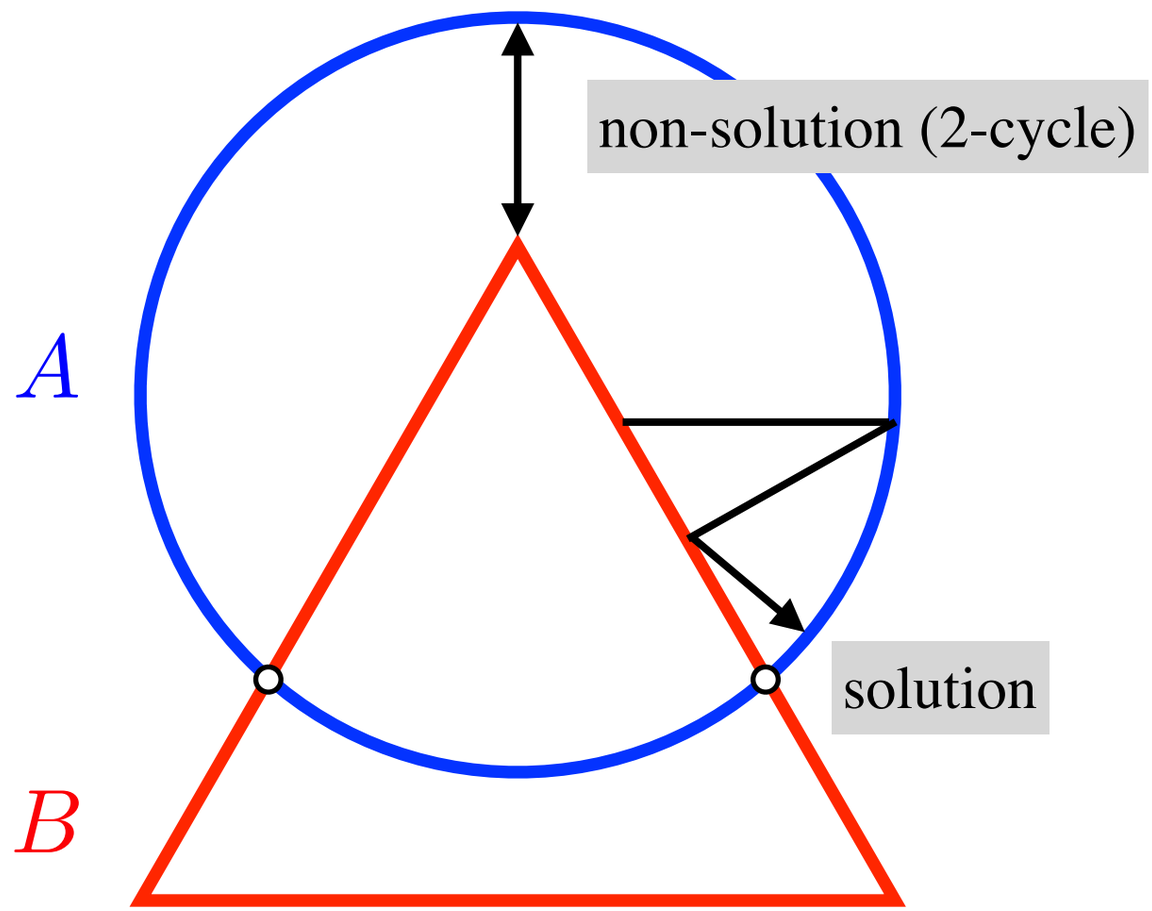
$$P_A(x) = \operatorname*{argmin}_{y \in A} \|y - x\|$$

the easy case …

$$A, B \text{ both convex}$$

$$\text{iterate: } x \to P_A(P_B(x))$$

non-convex A, B :



relaxed-reflect-reflect (RRR) iteration:

$$x \to (1 - \beta/2)x + (\beta/2)\, R_B(R_A(x))$$

$$R_A(x) = 2P_A(x) - x$$

$$\text{relaxation parameter: } 0 < \beta < 2$$

$$\text{“Douglas-Rachford”: } \beta = 1$$

Jon Borwein: “reflect-reflect-average”

RRR as dynamical system:

$$x \to x + \beta \left(P_B(2P_A(x) - x) - P_A(x) \right)$$

$$\beta : \text{ time-step}$$

$$P_B(2P_A(x) - x) - P_A(x) \neq \text{ gradient of a function}$$

(proof: periodic orbits for special A, B)

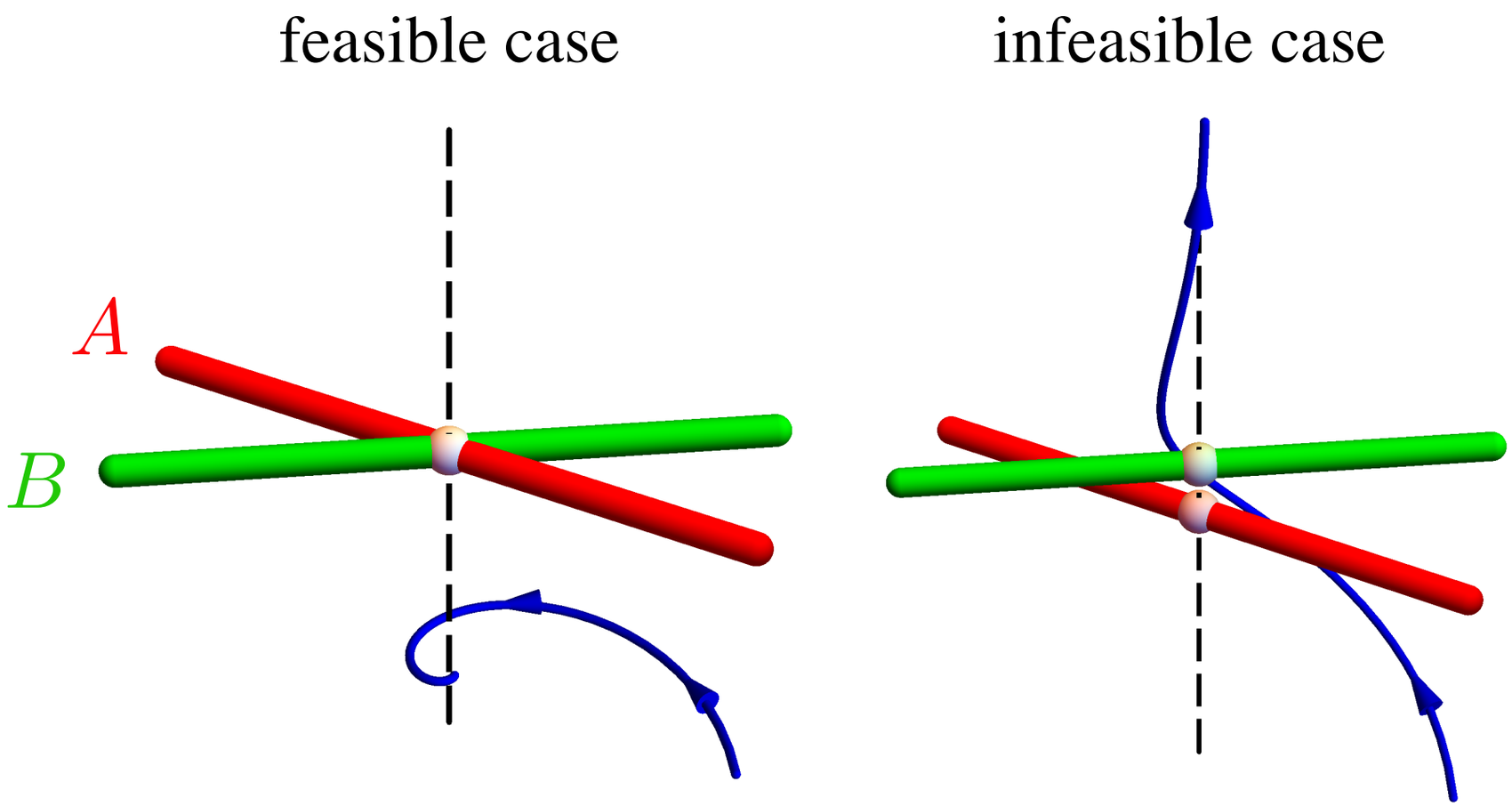
fixed points:

$$x^* = x^* + \beta \left(P_B(2P_A(x^*) - x^*) - P_A(x^*) \right)$$

$$P_B(2P_A(x^*) - x^*) = P_A(x^*) = x_{\text{sol}} \in A \cap B$$

in general, $x^* \notin A \cap B$

flow limit, $\beta \to 0$:



- exponentially contracting behavior in linear span of A, B
- uniform velocity v in orthogonal complement
- $v = 0$ in feasible case (linear space of fixed points)

a generalization …

$$P_B(2P_A(x) - x) - P_A(x)$$

$$\Big\downarrow \; \gamma > 0$$

$$P_B((1+\gamma)P_A(x) - \gamma\, x) - P_A((1-\gamma)P_B(x) + \gamma\, x)$$

a close relative: indicator-function ADMM*

$$y \leftarrow P_A(x+u)$$

$$x \leftarrow P_B(y-u)$$

$$u \leftarrow u + \alpha(x-y)$$

ADMM relaxation parameter: $0 < \alpha < 2$

$$u^* \text{ fixed point} \quad \Rightarrow \quad x = y \in A \cap B$$

$\alpha = 1$ ADMM and $\beta = 1$ RRR are equivalent

* Alternating Direction Method of Multipliers

three independent origins

- splitting method for PDEs
  Douglas-Rachford (DR) splitting, 1956

- convex optimization
  alternating direction method of multipliers (ADMM), 1974

- phase retrieval
  Fienup's hybrid input-output (HIO), 1982

a sampling of applications

shape of x	application
vector	bit retrieval
matrix	sudoku
tensor	pizza partitions
compound	machine learning

bit retrieval* :

reconstruct a ± 1 vector from its cyclic autocorrelations

example: $a(x), b(x) \in \mathbb{Z}[x]/(x^7 - 1)$

$$b(x) = 1 - x + x^2 + x^3 - x^4 + x^5 - x^6$$

$$\begin{aligned} a(x) &= b(x)b(1/x) \\ &= 7 - 5x + 3x^2 - x^3 - x^4 + 3x^5 - 5x^6 \end{aligned}$$

$$\text{coefficients} \equiv 3 \pmod 4$$

* phase retrieval for periodic, two-valued contrast

without noise, sub-exponential algorithm, $e^{c\,(N \log N)^{1/3}}$

N. Howgrave-Graham and M. Szydlo, *A method to solve cyclotomic norm equations* (2004)

with noise, only exponential algorithms

$$a(x) \to a(x) \pm 2(x + x^6) \pm 2(x^2 + x^5) \pm 2(x^3 + x^4)$$

RRR tolerates noise, finds proximal points on A, B

P_A : round to ± 1

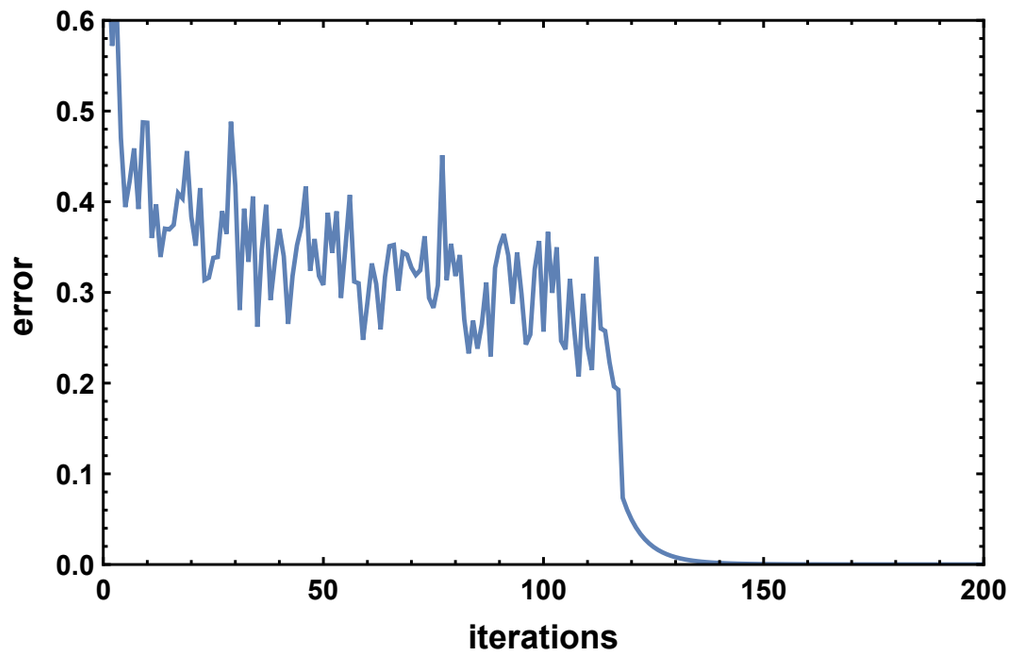
P_B : 1) FFT

2) rescale magnitudes*

3) FFT^{-1}

* to be consistent with noisy autocorrelations

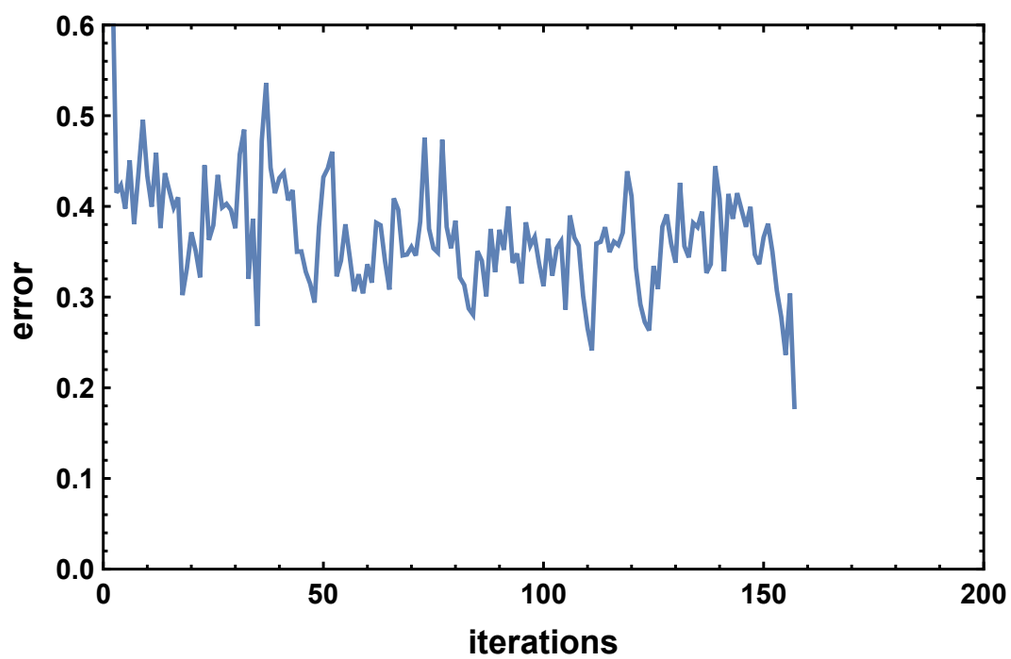
$N = 59$ bit instance



$$\text{error} = \|P_B(\cdots) - P_A(\cdots)\|$$

without noise:

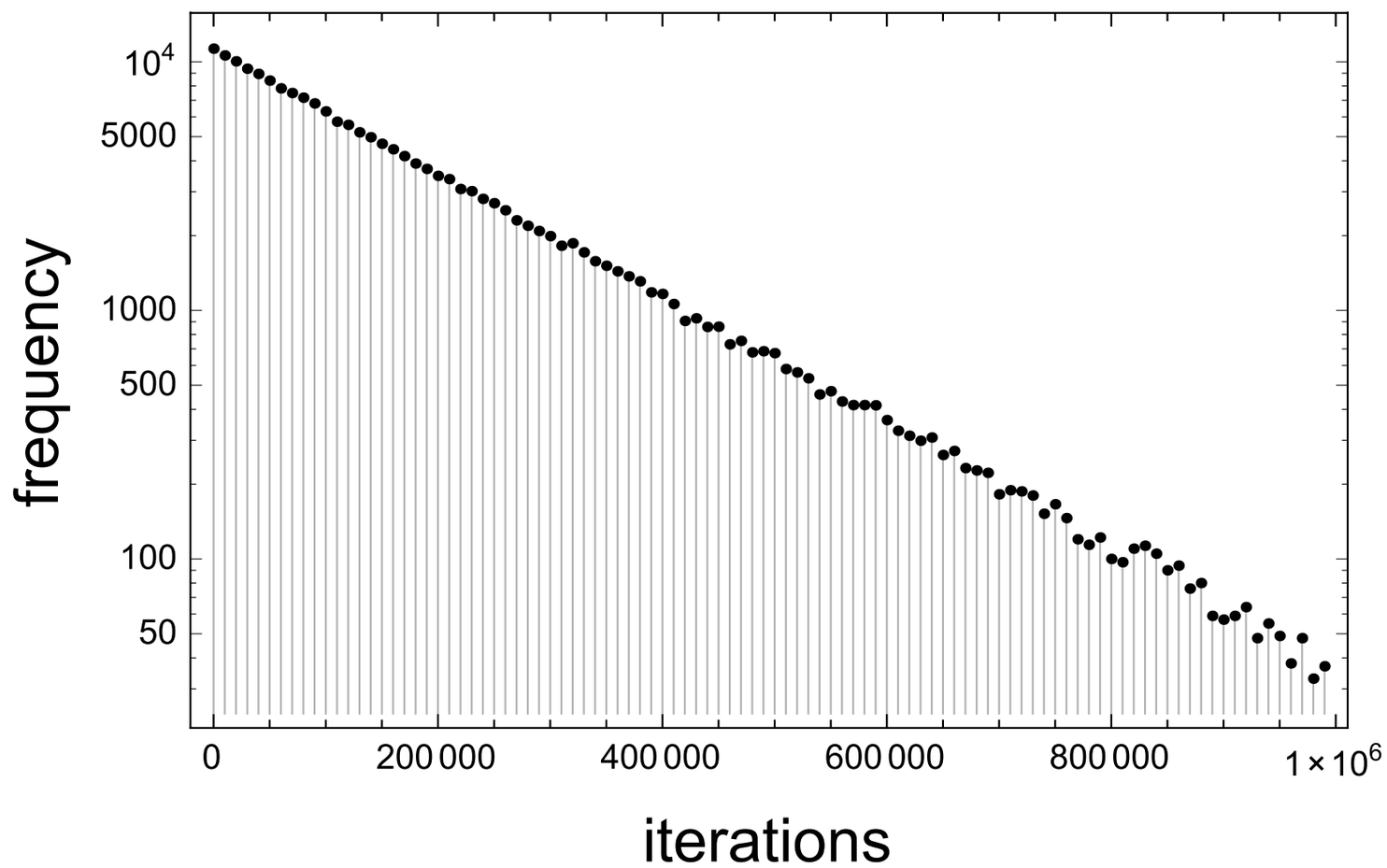
reconstructs N signs
(up to shift and reversal)



with noise:

terminate when all auto. off by ± 2
reconstructs N signs
and $(N - 1)/2$ noise bits!

distribution of iterations/run-times



200,000 solutions of $N = 43$ Hadamard instance (random starts)

$$a(x) = 43 - x - x^2 - x^3 - \cdots - x^{42}$$

RRR in the combinatorial setting

some bad questions:

- Is there a guarantee of convergence?
- What is the rate of convergence?

some good questions:

- Is the dynamics ergodic?
- What is the Poincaré recurrence time?

apropos ergodicity …

Does RRR solve any **sudoku**, given sufficient time?

Although "traps" in the usual sense (fixed points) are not a problem (they imply solutions), RRR can become trapped on limit cycles!

Until recently, all known sudoku formulations for RRR (sets A and B) exhibited limit cycle behavior in select puzzles (also judged hardest for humans).

success rates of hardest puzzles in “divide & concur” formulation

1	2		3					4
3	5					1		
		4						
		5	4			2		
6				7				
					8		9	
		3	1			5		
					9		7	
				6				8

red_dwarf 77.5%

							1	2
								3
		2	3			4		
		1	8					5
	6			7		8		
					9			
		8	5					
9				4		5		
4	7				6			

platinum_blonde 70%

								8
		3				4		
	9			2			6	
				7	9			
				6	1	2		
	6		5		2		7	
		8				5		
	1						2	
4		5						3

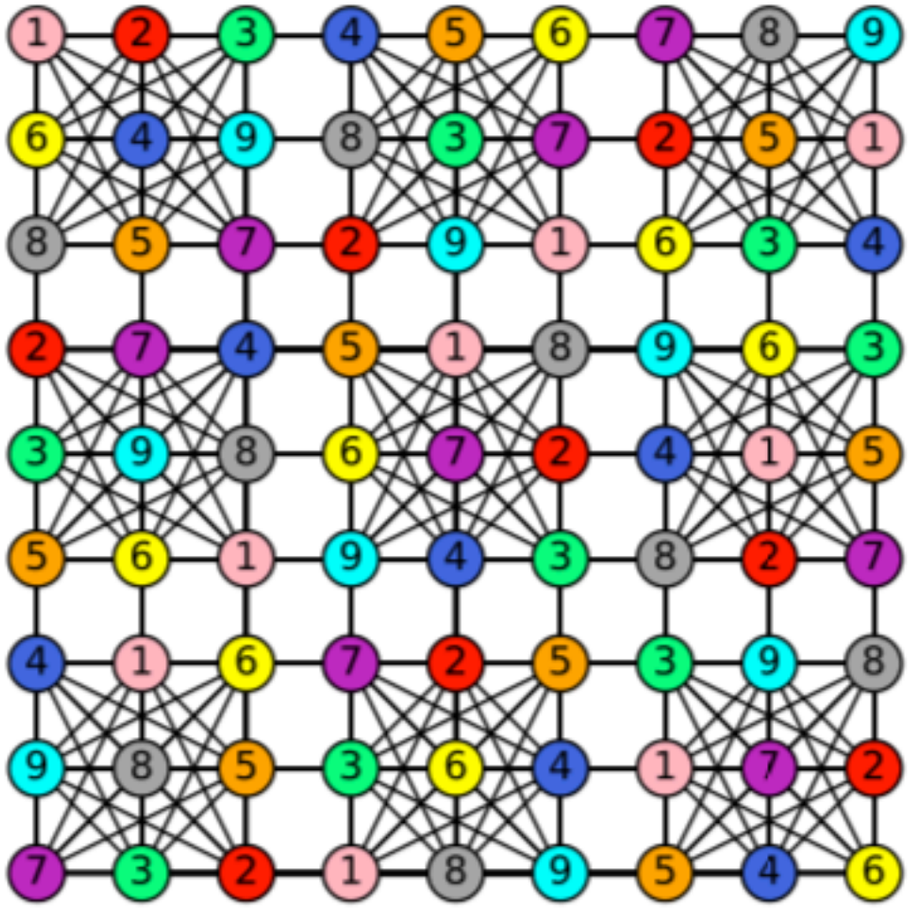
tarx0134 30%

				9			5	
	1						3	
		2	3			7		
		4	5				7	
8						2		
					6	4		
	9			1				
	8			6				
		5	4					7

coly013 0.48%

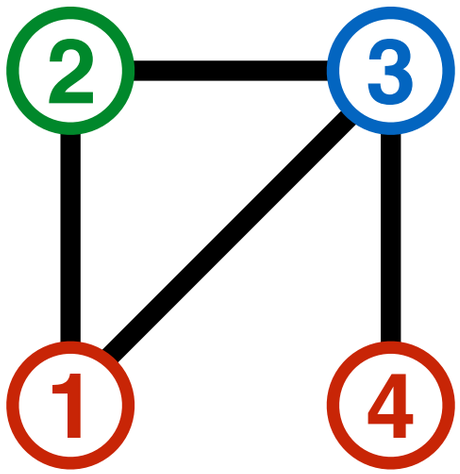
candidate ergodic formulation
step 1: graph coloring

1						7		9
	4				7	2		
8								
	7			1			6	
3								5
	6			4			2	
								8
		5	3				7	
7		2					4	6



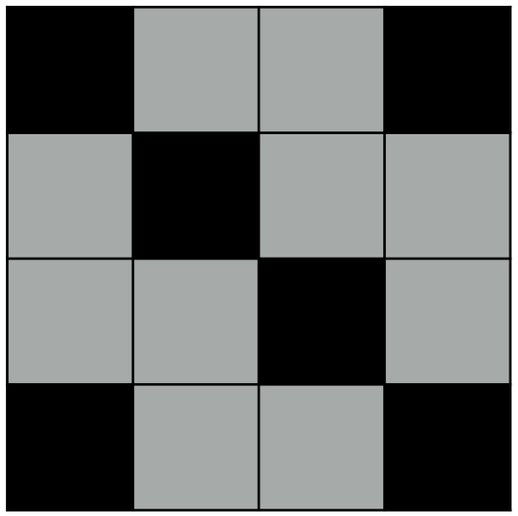
9-colored sudoku graph

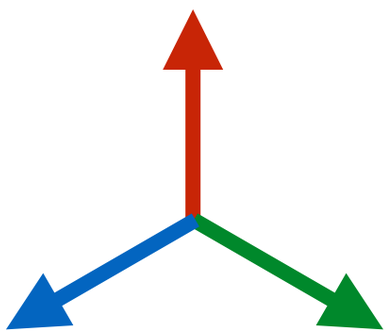
step 2: graph coloring as a constrained **matrix**



$$X_{ij} \in \begin{cases} \{-1/(\chi-1)\} & (i,j) \in E \\ \{-1/(\chi-1), 1\} & (i,j) \notin E \end{cases}$$

$$X_{ii} = 1$$





$$X_{12} = v_1 \cdot v_2$$

$\chi = 3$

X_{ij} known when colors i, j are given

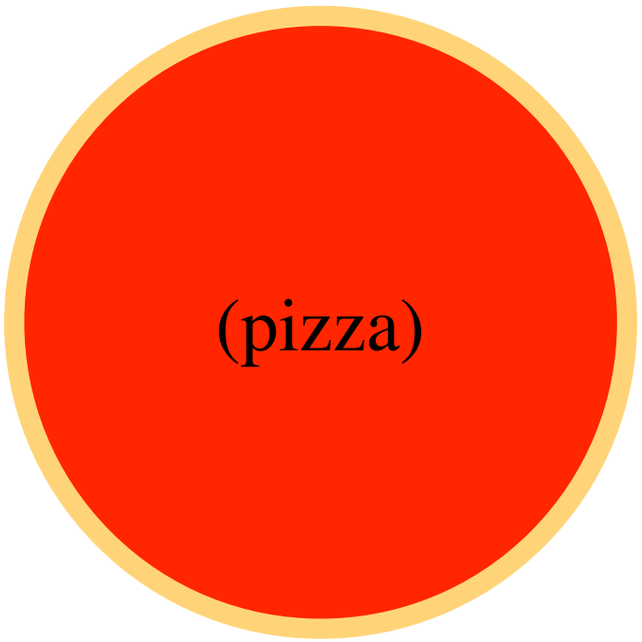
$P_A(X)$: element-wise, two-valued projection

$P_B(X)$: projection to nearest rank-$(\chi - 1)$ matrix

comparison with other $N \times N$ sudoku formulations

shape	A	B
$N \times N \times N$	$(N \times N \text{ permutation matrices})^{\times N}$ algorithm: "Hungarian"	
$4 \times N \times N \times N$	$(\text{1-hot})^{\times 4N^2}$	$(\text{concur})^{\times N^3}$
$N^2 \times N^2$	$\text{rank} = N - 1$ algorithm: eigen-decomposition	$\left\{1, \dfrac{-1}{N-1}\right\}^{\times N^4}$

pizza partitions



You have invited three people for pizza. They may not all show up.

Into how few pieces can you divide the pizza, in advance, so no matter how many guests show up the pieces can always be combined to give everyone an equal share?

3×4 solution tensor:

3	0	1	0	4
0	3	0	1	4
0	0	2	2	4
3	3	3	3	

pizza area = 12

6 pieces

for N people (max) seek:

$$N \times (N-1) \times \cdots \times \lfloor N/2 + 1 \rfloor \text{ tensor with properties}$$

A : equipartitioned in each dimension

B : k nonzeros

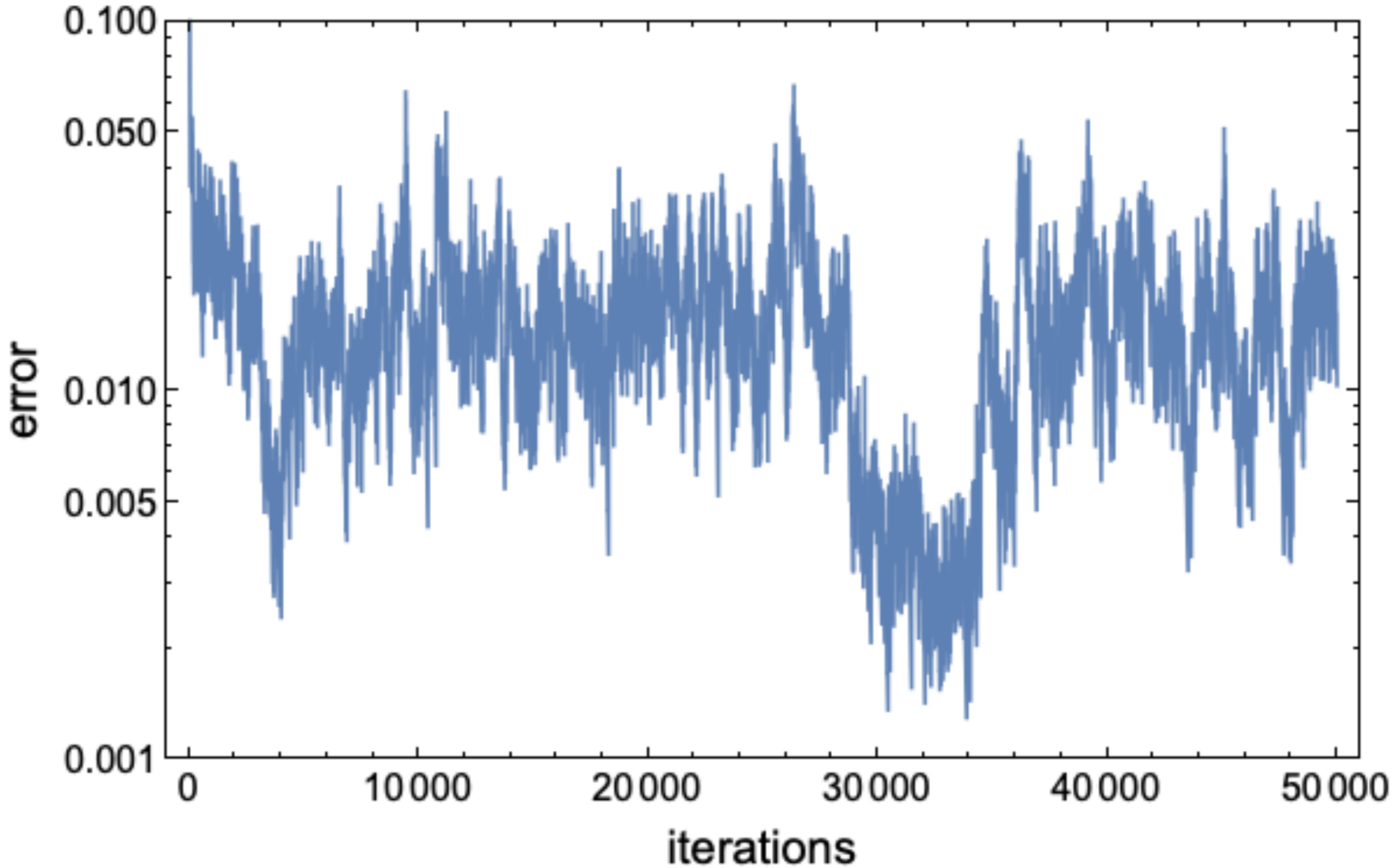
P_A :

1. In any dimension, apply shift to each layer (row, column, etc.) that gives correct sum.

2. Do this to all dimensions in any order (they commute).

P_B : Zero negative elements, sort positives, leave the k largest unchanged, and zero the rest.

7 pizza consumers, 14 pieces $(4 \times 5 \times 6 \times 7)$



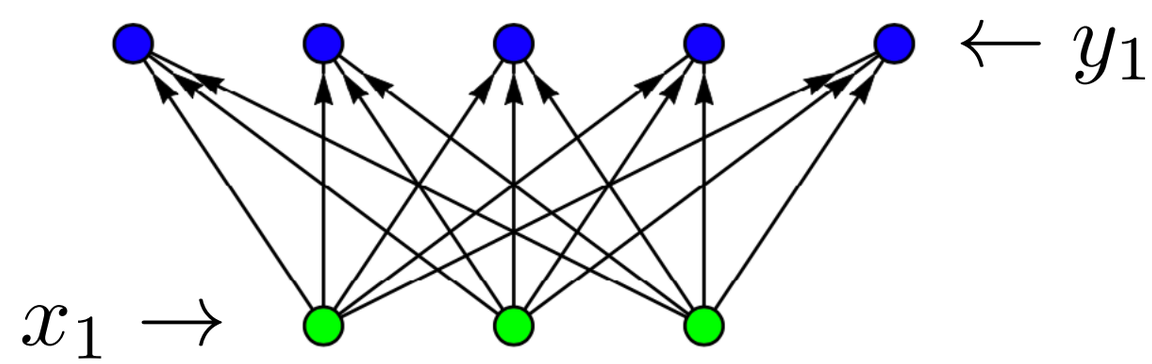
10^8 iterations later ...

solution: `5, 6, 11, 15, 21, 25, 29, 31, 35, 39, 45, 49, 54, 55`

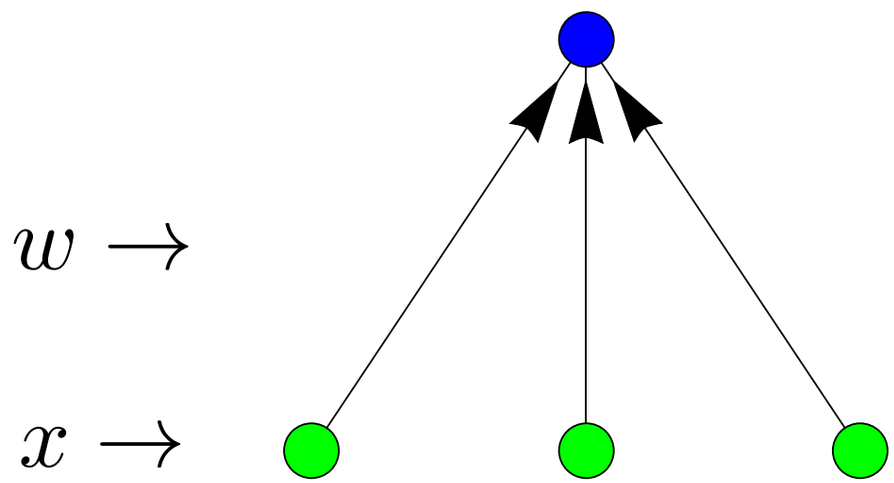
non-negative matrix factorization

$$XW = Y$$

k instantiations of neural net, representing non-negative data in terms of non-negative codes and weights



neuron (bilinear) constraint



for arbitrary x, w (code vector, incident weights)

$$\text{minimize:} \quad \|x' - x\|^2 + \|w' - w\|^2$$

$$\text{such that:} \quad x' \cdot w' = y$$

unique whenever $x \pm w \neq 0$

divide & concur trick

copies of weights for each network instantiation:

$$w[i \to j] \to w[i \to j, c] \ , \quad c = 1, 2, \ldots, k$$

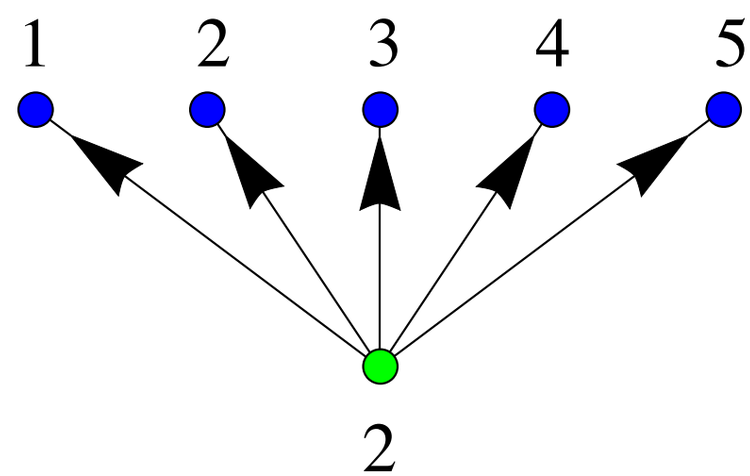
copies of code node for each receiving data node:

$$x[i, c] \to x[i \to j, c] \ , \quad j = 1, 2, \ldots, \#(\text{data nodes})$$

local neuron constraint:

$$\sum_i x[i \to j, c] \ w[i \to j, c] = y[j, c]$$

A: concur, non-negatively, on weights
concur, non-negatively, on codes



projection:

replace $x[2 \to 1], x[2 \to 2], x[2 \to 3], x[2 \to 4], x[2 \to 5]$ by their average or zero when average is negative

B: local neuron constraints

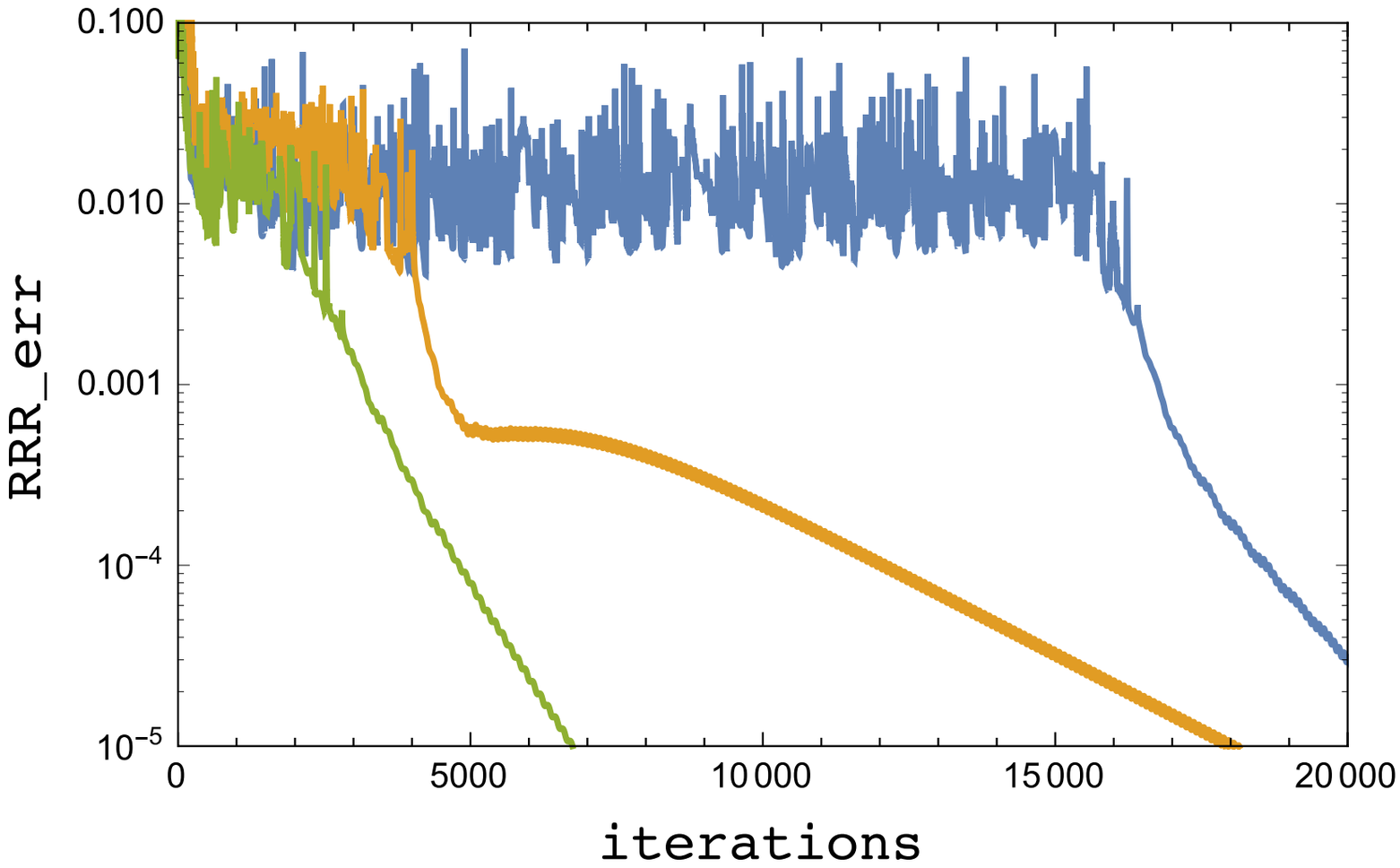
linear Euclidean distance matrices (LEDMs)

$$X \qquad\qquad W \qquad\qquad Y_{ij} = (i-j)^2$$

$$\begin{pmatrix} 0 & 5 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 3 & 1 \\ 0 & 0 & 1 & 5 & 0 \end{pmatrix} \begin{pmatrix} 4 & 1 & 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 1 & 3 & 5 \\ 0 & 1 & 4 & 4 & 1 & 0 \\ 5 & 3 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 4 & 9 & 16 & 25 \\ 1 & 0 & 1 & 4 & 9 & 16 \\ 4 & 1 & 0 & 1 & 4 & 9 \\ 9 & 4 & 1 & 0 & 1 & 4 \\ 16 & 9 & 4 & 1 & 0 & 1 \\ 25 & 16 & 9 & 4 & 1 & 0 \end{pmatrix}$$

non-negative rank $r_+ = 5$ ordinary rank $= 3$

6×6 LEDM



$n \times n$ LEDMs

		RRR		coordinate descent*	
n	r_+	rate	sec	rate	sec
6	5	100%	2	1.6%	100
8	6	100%	59	< 1%	—
12	7	100%	567	< 1%	—
16	8	100%	3677	< 1%	—

* scikit-learn

from *Learning internal representations by error propagation*

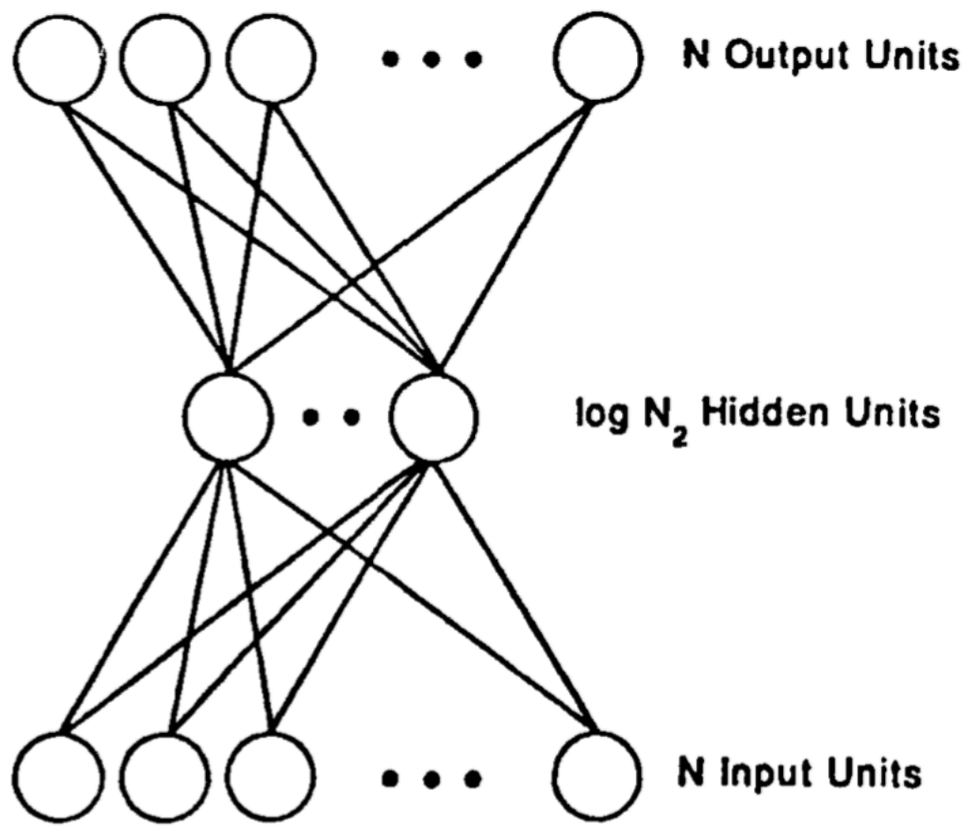


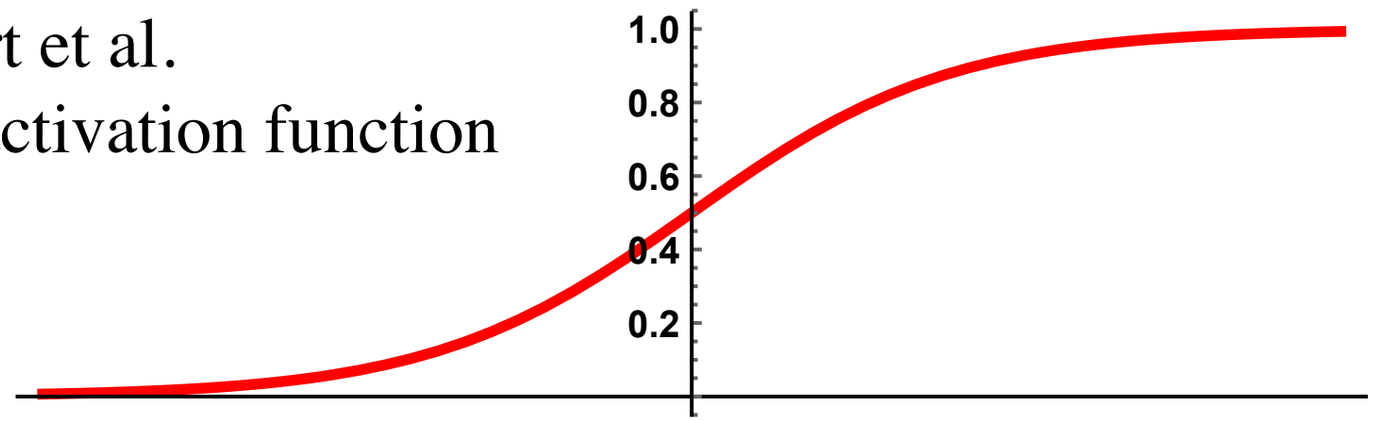
FIGURE 7. A network for solving the encoder problem. In this problem there are N orthogonal input patterns each paired with one of N orthogonal output patterns. There are only $\log N_2$ hidden units. Thus, if the hidden units take on binary values, the hidden units must form a binary number to encode each of the input patterns. This is exactly what the system learns to do.

The Encoding Problem

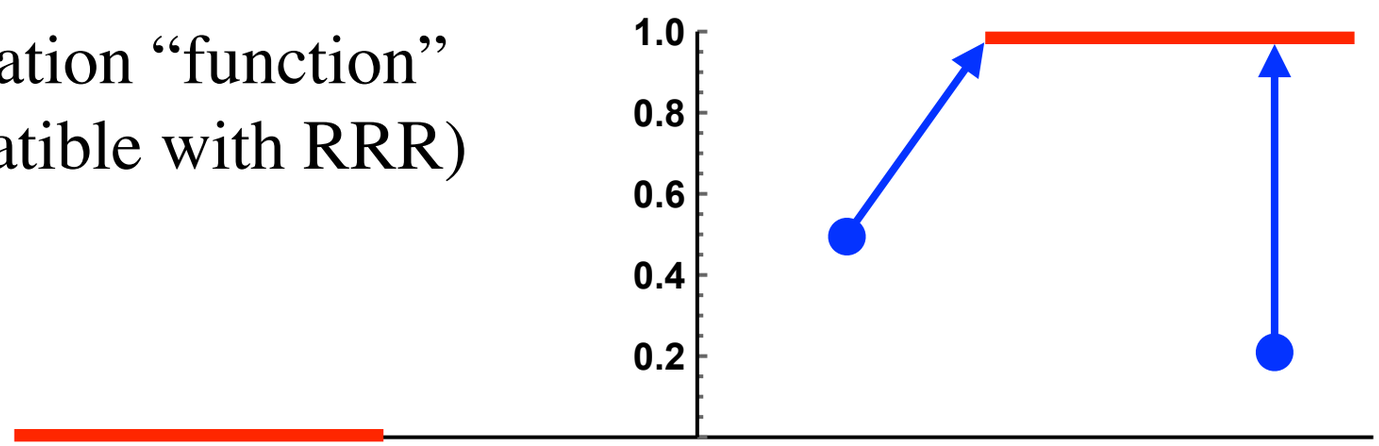
Input Patterns		Hidden Unit Patterns				Output Patterns
10000000	→	.5	0	0	→	10000000
01000000	→	0	1	0	→	01000000
00100000	→	1	1	0	→	00100000
00010000	→	1	1	1	→	00010000
00001000	→	0	1	1	→	00001000
00000100	→	.5	0	1	→	00000100
00000010	→	1	0	.5	→	00000010
00000001	→	0	0	.5	→	00000001

cheating!

Rumelhart et al.
sigmoid activation function



step activation "function"
(compatible with RRR)



RRR "discovered" proper binary encoding up to 2^5

Reasons to consider RRR for combinatorial problems:

- sufficiently different from standard search heuristics
- able to leverage powerful algorithms as projections
- easy concurrency (independent local constraints)
- easy parallelism (random starts)
- easy to program!

Further reading

Elser, Veit, I. Rankenburg, and P. Thibault. "Searching with iterated maps." *Proceedings of the National Academy of Sciences* 104.2 (2007): 418-423.

Gravel, Simon, and Veit Elser. "Divide and concur: A general approach to constraint satisfaction." *Physical Review E* 78.3 (2008): 036706.

Elser, Veit. "The complexity of bit retrieval." *IEEE Transactions on Information Theory* 64.1 (2017): 412-428.

Lindstrom, Scott B., and Brailey Sims. "Survey: Sixty Years of Douglas-Rachford." *arXiv preprint arXiv:1809.07181* (2018).

Artacho, F. J., Rubén Campoy, and Veit Elser. "An enhanced formulation for solving graph coloring problems with the Douglas-Rachford algorithm." *arXiv preprint arXiv:1808.01022* (2018).