10-3-2015

Minimum Richness Equilibrium and Sudoku

Jeffrey Horn
Northern Michigan University, jhorn@nmu.edu

Minimum Richness Equilibrium and Sudoku

Jeffrey Horn

Northern Michigan University

Department of Mathematics and Computer Science

Marquette, MI USA

jhorn@nmu.edu

http://cs.nmu.edu/~jeffhorn

MAA U.P. Regional Meeting

October 3, 2015

Lake Superior State University

Abstract

In the field of theoretical ecology the term "richness" refers to the number of species present in an ecosystem. By reducing the game of Sudoku to the problem of exact four cover (X4C), then reducing X4C to minimum richness equilibrium (MRE), we show that MRE is in NP-complete. We further reduce MRE to minimum weight linear programming (MWLP) to arrive at a simple, polynomial-time decision process that we demonstrate to be a pretty darn good Sudoku solver!

Background: Genetic Algorithms

The simple GA:

p_x is the proportion of individual/species x in the population.
f_x is the fitness of x.

Proportionate selection:

$$p_x(t+1) = p_x(t)\,\frac{f_x}{\bar{f}}$$

Example:

$$p_A(t+1) = p_A(t)\,\frac{f_A}{p_A(t)\,f_A + p_B(t)\,f_B}$$

Background: The RFS Approach

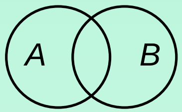
The **SHARED FITNESS** $f_{sh,x}$ of a species x depends, in a simple way, on competition from overlapping species:

$$f_{sh,x} = \frac{1}{niche\_count(x)} \quad \longleftrightarrow \quad = \frac{1}{\sum_{\text{all species } y} p_y f_{xy}}$$

where p_y is the **proportion** of species ***y*** in the current population.

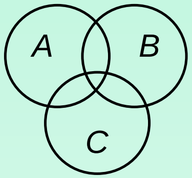
Example for two overlapping niches

$$f_{Sh,A} = \frac{f_A}{n_A f_A + n_B f_{AB}}$$



Example for three overlapping niches

$$f_{Sh,A} = \frac{f_A}{n_A f_A + n_B f_{AB} + n_C f_{AC}}$$



Finally, a **selection operator**, such as *proportionate selection*, uses the RFS Shared Fitnesses each generation.

- Shared Fitness: $f_{sh}(x) = \left(\sum_{y \in S} p_y * f_{x,y}\right)^{-1}$

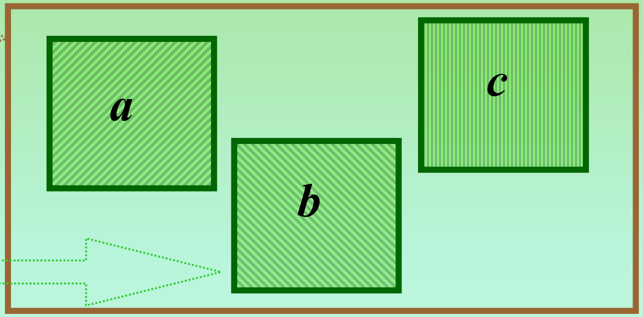
Background: The RFS Approach

Resource-defined Fitness Sharing (RFS)
introduced by Horn (2002) as a **synthesis**
of Fitness Sharing and Resource Sharing.

Substrate (stock material)

is a finite **RESOURCE** to be **COVERED**

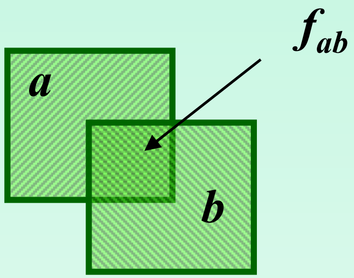
by niches (defined by the species).



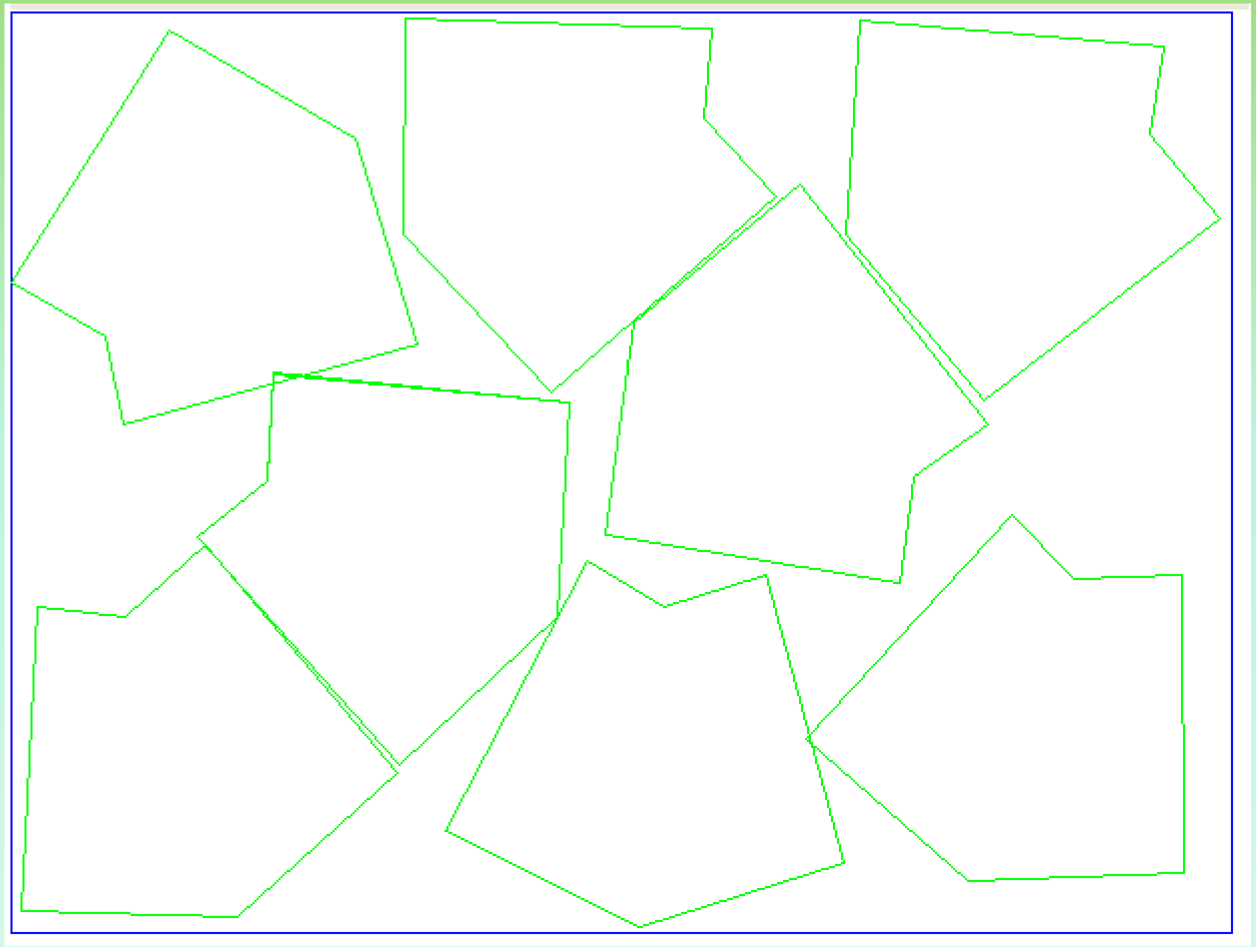
Each **SPECIES** covers a unique subset of the resources.

Overlapping species compete for the shared amount of resource.

E.g., Species a and b overlap
in coverage by amount f_{ab} :

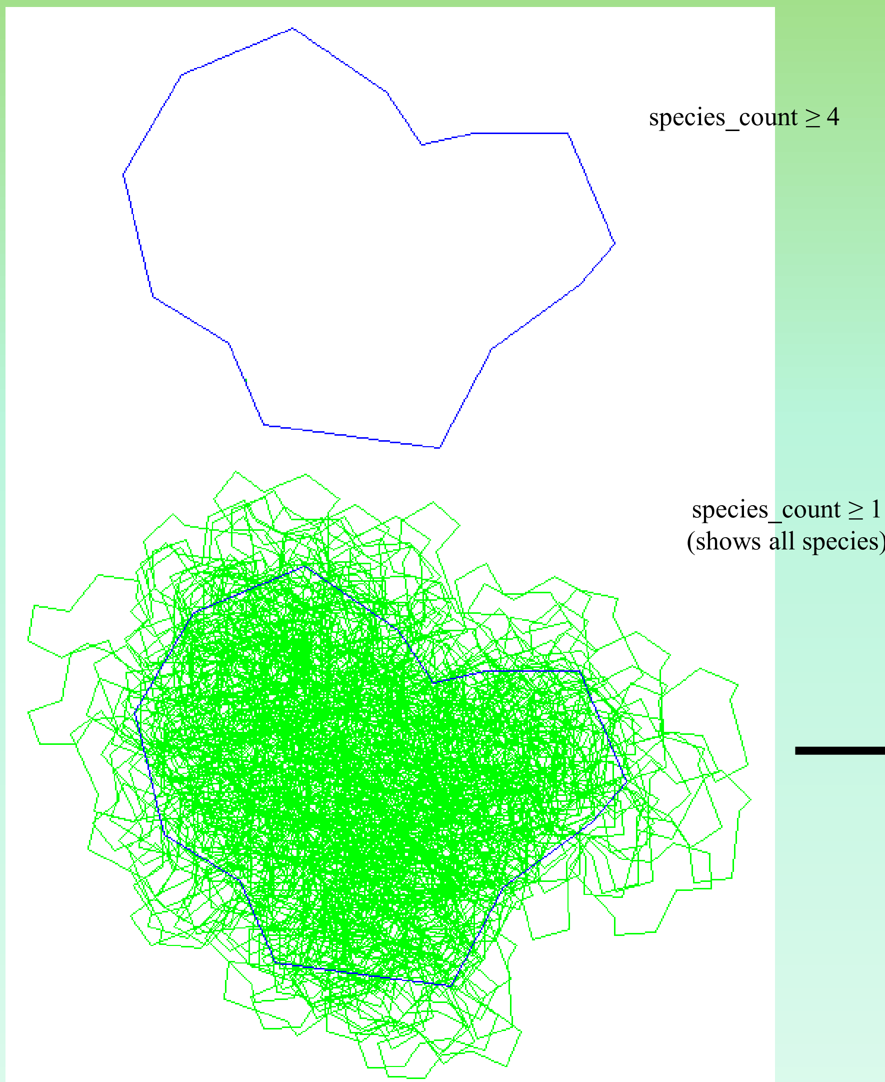


RFS for Shape Nesting

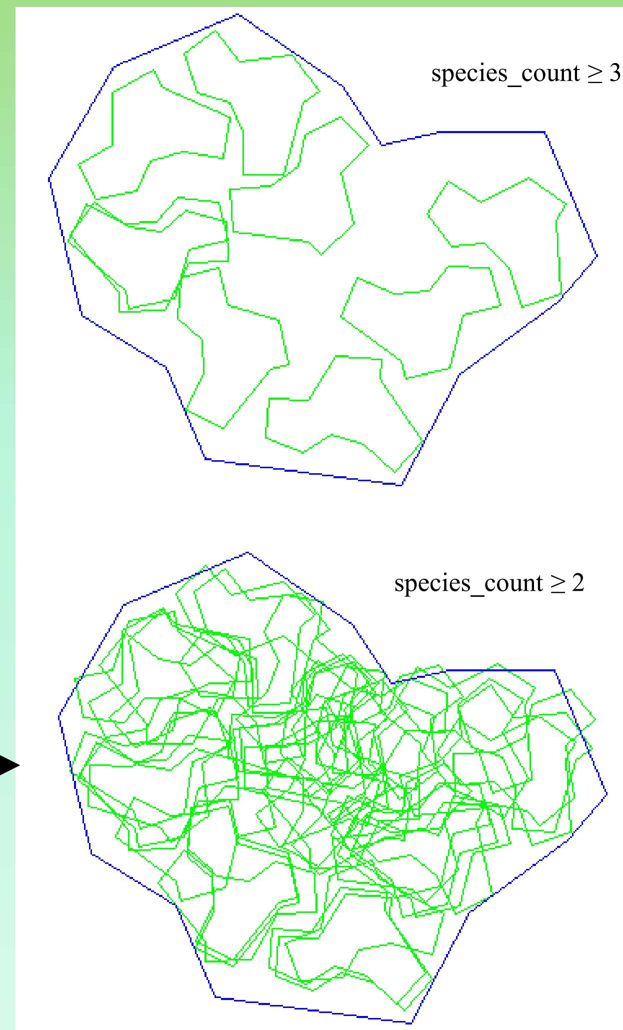


Experiment 1

generation 1
(one generation beyond the initial population)

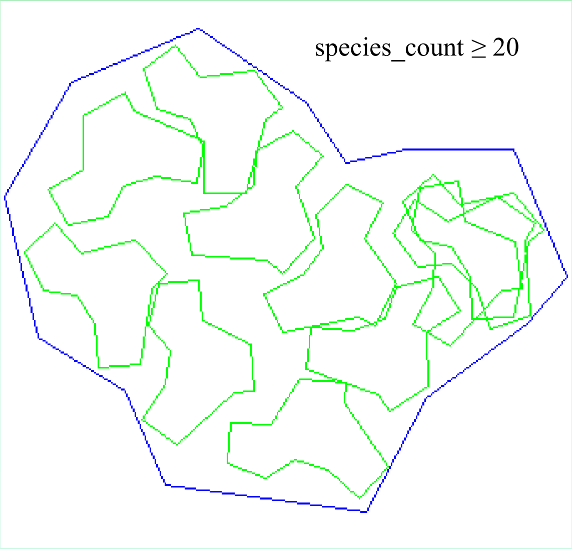


generation 209
(8 cooperative species)

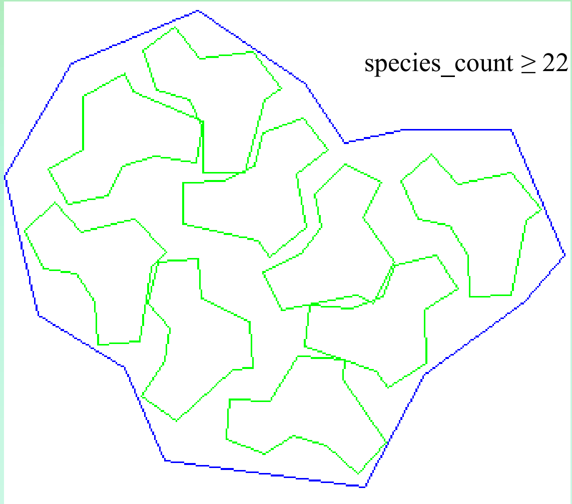


Experiment 1

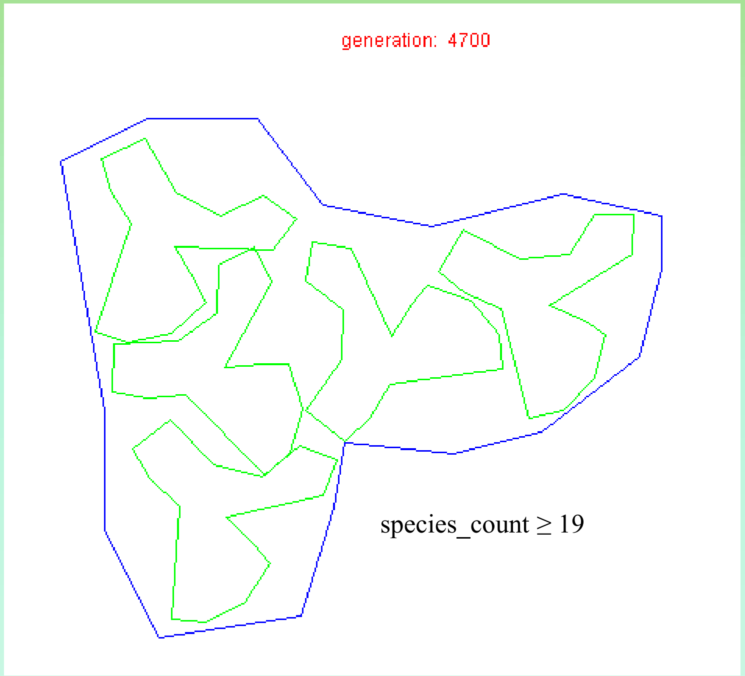
generation 609
(almost 9 cooperative species)



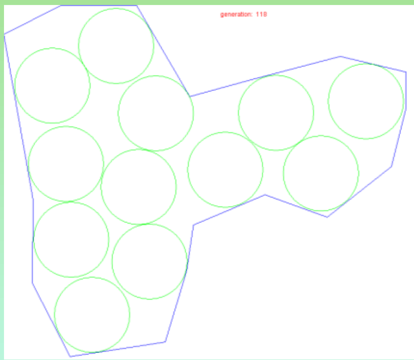
generation 709
(9 cooperative species)

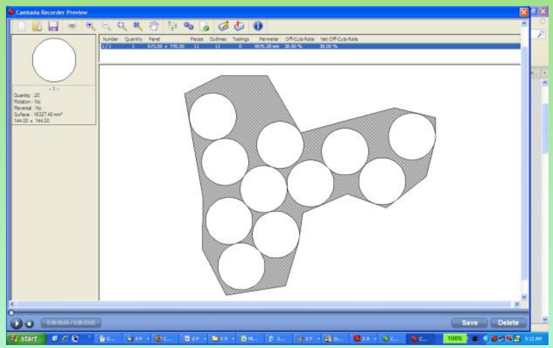


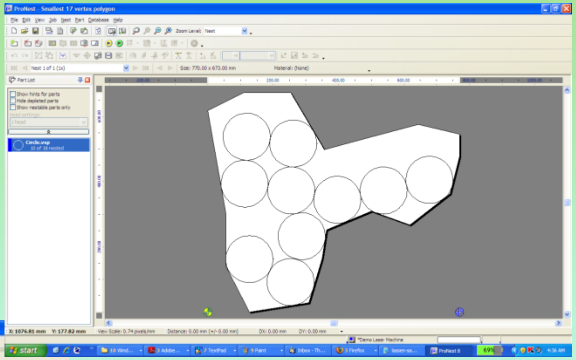
Experiment 2

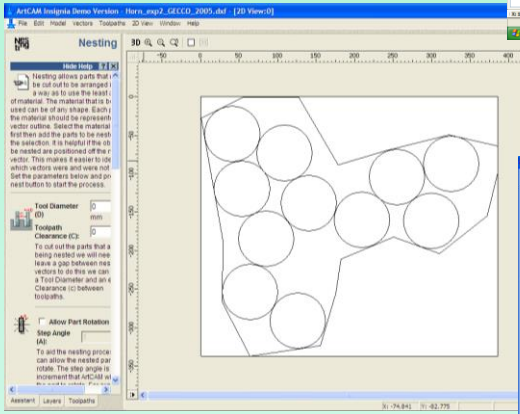


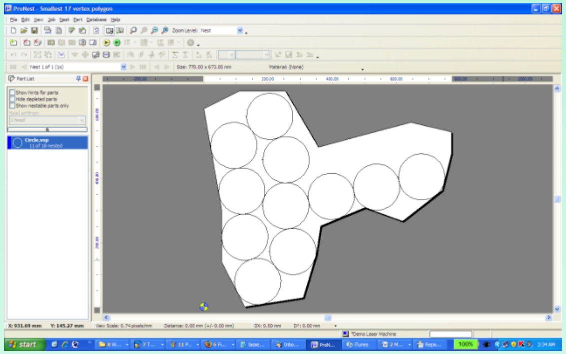
Co-evolutionary Shape Nesting can Outperform Commercial Software

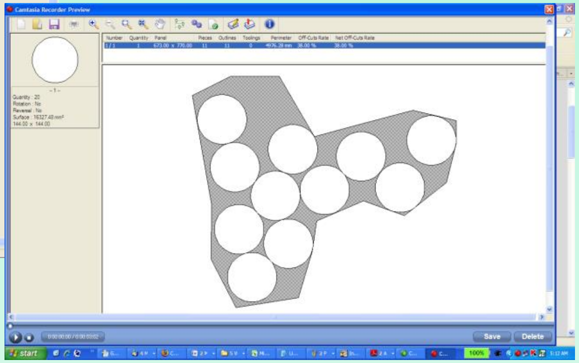




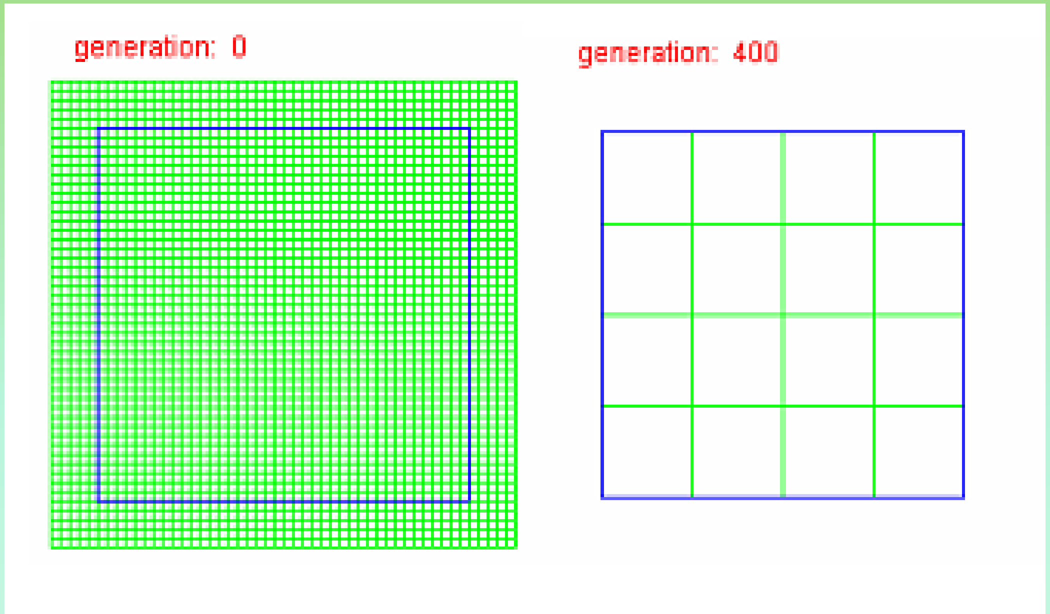






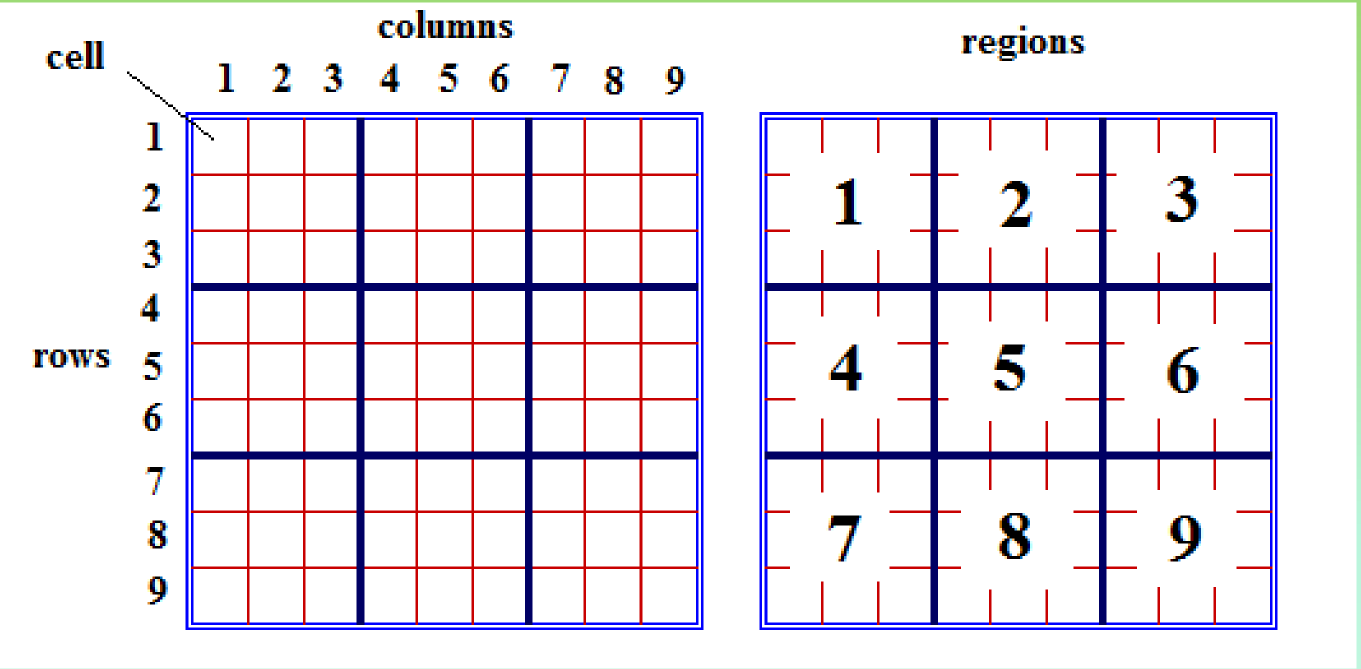


Goal: Evolve an Exact Cover of Substrate by *K* Species



Here K = 16.

9x9 Sudoku



Insert the numerals 1-9 in each cell subject to FOUR constraints:

1. Only one numeral per cell
2. Exactly one of each numeral per row
3. Exactly one of each numeral per column
4. Exactly one of each numeral per region

EASY Puzzle No. 67

clues

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

solution

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

HARD Puzzle No. 6

clues

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

solution

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

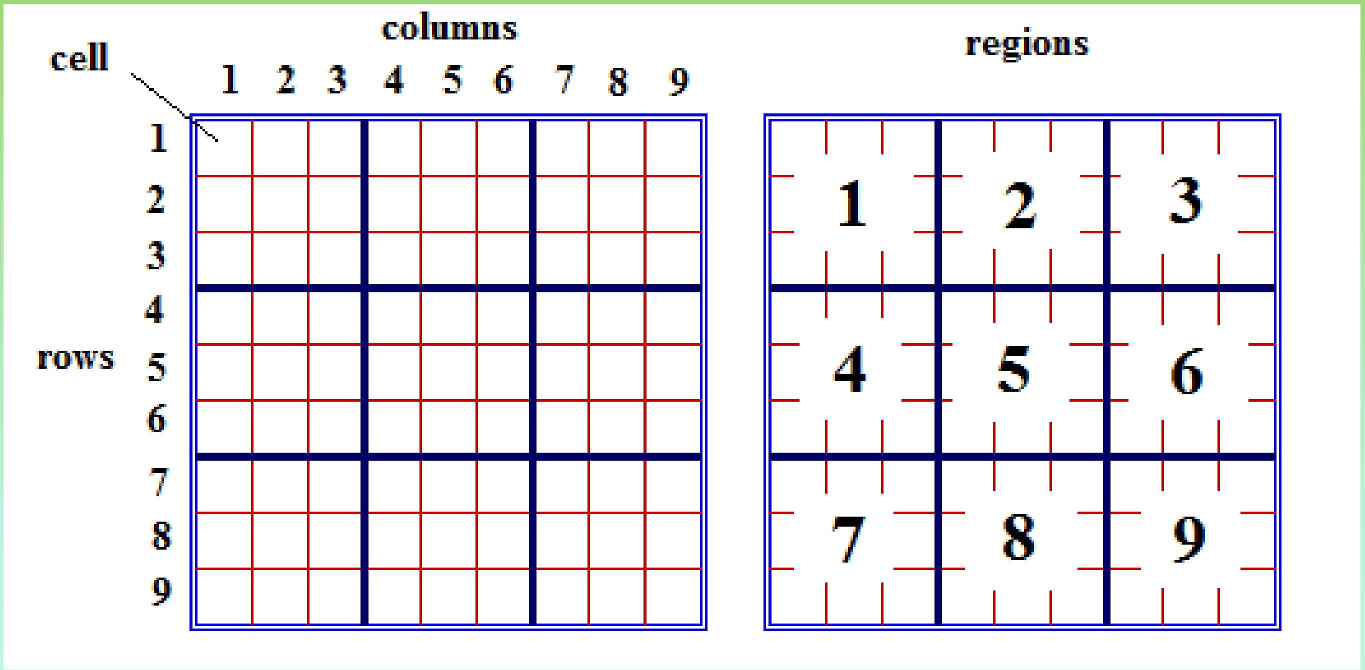
Garey and Johnson (1979): X3C

[SP2] **EXACT COVER BY 3-SETS (X3C)**

INSTANCE: Set X with $|X| = 3q$ and a collection C of 3-element subsets of X.
QUESTION: Does C contain an exact cover for X, i.e., a subcollection $C' \subseteq C$ such that every element of X occurs in exactly one member of C'?

Reference: [Karp, 1972]. Transformation from 3DM.
Comment: Remains NP-complete if no element occurs in more than three subsets, but is solvable in ploynomial time if no element occurs in more than two subsets [Garey and Johnson, ——]. Related EXACT COVER BY 2-SETS problem is also solvable in polynomial time by matching techniques.

9x9 Sudoku



Insert the numerals 1-9 in each cell subject to FOUR constraints:

1. Only one numeral per cell
2. Exactly one of each numeral per row
3. Exactly one of each numeral per column
4. Exactly one of each numeral per region

Sudoku as X4C

$$f_{x,y} = 0 + \left\{ \begin{array}{lll} 1/4 & \text{iff} & \text{SameCell}(x,y) \\ 1/4 & \text{iff} & \text{SameNumeral}(x,y) \wedge \text{SameRow}(x,y) \\ 1/4 & \text{iff} & \text{SameNumeral}(x,y) \wedge \text{SameColumn}(x,y) \\ 1/4 & \text{iff} & \text{SameNumeral}(x,y) \wedge \text{SameRegion}(x,y) \end{array} \right\} \text{(cumulative)},$$

The RFSS-Evolve Algorithm

1. INITIALIZE:
   (a) Generate initial set of species (unique chromosomes) S;
   (b) $\forall x, y \in S$: calculate pairwise intersection[3] and store as $f_{x,y} := \frac{|x \cap y|}{|x|}$;
   (c) $\forall s \in S : p_s := \frac{1}{|S|}$; // Uniform distribution across all species, initially.
2. LOOP: while (termination condition[4] is false) do

2. LOOP: while (termination condition[4] is false) do

   (a) $\forall x \in S$: Evaluate and store shared fitnesses as $f_{sh}(x) := (\sum_{\forall y \in S} p_y * f_{x,y})^{-1}$;

   (b) Calculate and store average shared fitness as $\overline{f_{sh}} := \sum_{\forall x \in S}(p_x * f_{sh}(x))$;

   (c) Calculate next generation species proportions p' as $\forall x \in S : p'_x := p_u * \frac{f_{sh}(x)}{\overline{f_{sh}}}$;
   // This implements proportionate selection, using shared fitnesses.

   (d) Move to next generation by updating species proportions as $\forall x \in S : p_x := p'_x$;

Table 2: Raw Results from Runs on USA Today 200 Sudoku Puzzle Book

PUZZLE No. (within level)	DIFFICULTY LEVEL			PUZZLE No. (within level)	DIFFICULTY LEVEL		
	EASY	*MEDIUM*	*HARD*		*EASY*	*MEDIUM*	*HARD*
1	580	432	2622	35	291	1154	6187
2	146	2094	3596	36	125	1276	4131
3	90	365	2028	37	697	913	Not Solved
4	323	326	1987	38	98	382	831
5	254	317	2059	39	166	416	3225
6	521	706	Not Solved	40	335	508	817
7	558	728	685	41	372	379	580
8	342	1103	3341	42	93	749	1751
9	134	905	Not Solved	43	216	1369	4526
10	285	361	Not Solved	44	312	490	Not Solved
11	79	766	911	45	297	266	Not Solved
12	72	141	859	46	34	164	1161
13	58	229	2611	47	133	1028	248
14	245	2829	Not Solved	48	162	180	968
15	344	922	3640	49	232	1535	689
16	356	280	Not Solved	50	121	492	1360
17	140	425	2397	51	400	616	1077
18	447	696	1044	52	80	471	Not Solved
19	513	489	775	53	802	844	676
20	242	279	392	54	185	250	617
21	638	634	3630	55	343	479	9891
22	255	435	1066	56	90	361	Not Solved
23	314	125	826	57	206	900	1156

Species Interaction Matrix

$\mathbf{M}_{RFS} =$ Matrix of pairwise overlaps between species

Under RFS: $\mathbf{M}_{RFS} * \overrightarrow{p} = \overrightarrow{niche\_count}$

vector of species proportions

vector of species niche counts

$$
\left[\begin{array}{cccc|cccc}
f_{E_1,E_1} & f_{E_1,E_2} & \cdots & f_{E_1,E_K} & f_{E_1,C_1} & f_{E_1,C_2} & \cdots & f_{E_1,C_H} \\
f_{E_2,E_1} & f_{E_2,E_2} & \cdots & f_{E_2,E_K} & f_{E_2,C_1} & f_{E_2,C_2} & \cdots & f_{E_2,C_H} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
f_{E_K,E_1} & f_{E_K,E_2} & \cdots & f_{E_K,E_K} & f_{E_K,C_1} & f_{E_K,C_2} & \cdots & f_{E_K,C_H} \\
\hline
f_{E_1,C_1} & f_{E_2,C_1} & \cdots & f_{E_K,C_1} & f_{C_1,C_1} & f_{C_1,C_2} & \cdots & f_{C_1,C_H} \\
f_{E_1,C_2} & f_{E_2,C_2} & \cdots & f_{E_K,C_2,} & f_{C_1,C_2} & f_{C_2,C_2} & \cdots & f_{C_2,C_H} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
f_{E_1,C_H} & f_{E_2,C_H} & \cdots & f_{E_K,C_H} & f_{C_1,C_H} & f_{C_2,C_H}, & \cdots & f_{C_H,C_H}
\end{array}\right]
\left[\begin{array}{c}
p_{E_1} \\ p_{E_2} \\ \vdots \\ p_{E_K} \\ \hline p_{C_1} \\ p_{C_2} \\ \vdots \\ p_{C_H}
\end{array}\right]
=
\left[\begin{array}{c}
niche\_count(E_1) \\ niche\_count(E_1) \\ \vdots \\ niche\_count(E_K) \\ niche\_count(C_1) \\ niche\_count(C_2) \\ \vdots \\ niche\_count(C_H)
\end{array}\right]
$$

Notation

Modified Notation:

$f_{E_i E_j}$ ≡ overlap between species E_i and species E_j

p_{E_2} ≡ proportion of population occupied by species E_2

Property I

$$f_{E_i E_j} = f_{E_j E_i} = 0, \text{ for } i \neq j$$

Exact Cover species do not compete! (no overlaps)

Property II

$$\forall_{i \in (1..k)} \sum_{j=1}^{h} f_{E_j C_i} = 1$$

Exact Cover species completely cover all other species

Species Interaction Matrix

By **Property I:**

$$f_{E_i E_j} = f_{E_j E_i} = 0, \text{ for } i \neq j$$

Note that on main diagonal of MRFS: $f_{xx} = 1$

$$\left[\begin{array}{cccc|cccc} f_{E_1,E_1} & f_{E_1,E_2} & \cdots & f_{E_1,E_K} & f_{E_1,C_1} & f_{E_1,C_2} & \cdots & f_{E_1,C_H} \\ f_{E_2,E_1} & f_{E_2,E_2} & \cdots & f_{E_2,E_K} & f_{E_2,C_1} & f_{E_2,C_2} & \cdots & f_{E_2,C_H} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{E_K,E_1} & f_{E_K,E_2} & \cdots & f_{E_K,E_K} & f_{E_K,C_1} & f_{E_K,C_2} & \cdots & f_{E_K,C_H} \\ \hline f_{E_1,C_1} & f_{E_2,C_1} & \cdots & f_{E_K,C_1} & f_{C_1,C_1} & f_{C_1,C_2} & \cdots & f_{C_1,C_H} \\ f_{E_1,C_2} & f_{E_2,C_2} & \cdots & f_{E_K,C_2}, & f_{C_1,C_2} & f_{C_2,C_2} & \cdots & f_{C_2,C_H} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{E_1,C_H} & f_{E_2,C_H} & \cdots & f_{E_K,C_H} & f_{C_1,C_H} & f_{C_2,C_H}, & \cdots & f_{C_H,C_H} \end{array}\right] \left[\begin{array}{c} p_{E_1} \\ p_{E_2} \\ \vdots \\ p_{E_K} \\ \hline p_{C_1} \\ p_{C_2} \\ \vdots \\ p_{C_H} \end{array}\right] = \left[\begin{array}{c} niche\_count(E_1) \\ niche\_count(E_1) \\ \vdots \\ niche\_count(E_K) \\ niche\_count(C_1) \\ niche\_count(C_2) \\ \vdots \\ niche\_count(C_H) \end{array}\right]$$

Species Interaction Matrix

$$
\left[\begin{array}{cccc|cccc}
1 & 0 & \dots & 0 & f_{E_1,C_1} & f_{E_1,C_2} & \cdots & f_{E_1,C_H} \\
0 & 1 & \dots & 0 & f_{E_2,C_1} & f_{E_2,C_2} & \cdots & f_{E_2,C_H} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \dots & 1 & f_{E_K,C_1} & f_{E_K,C_2} & \cdots & f_{E_K,C_H} \\
\hline
f_{E_1,C_1} & f_{E_2,C_1} & \cdots & f_{E_K,C_1} & 1 & f_{C_1,C_2} & \cdots & f_{C_1,C_H} \\
f_{E_1,C_2} & f_{E_2,C_2} & \cdots & f_{E_K,C_2} & f_{C_1,C_2} & 1 & \cdots & f_{C_2,C_H} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
f_{E_1,C_H} & f_{E_2,C_H} & \cdots & f_{E_K,C_H} & f_{C_1,C_H}, & f_{C_2,C_H} & \cdots & 1
\end{array}\right]
\left[\begin{array}{c}
p_{E_1} \\ p_{E_2} \\ \vdots \\ p_{E_K} \\ \hline p_{C_1} \\ p_{C_2} \\ \vdots \\ p_{C_H}
\end{array}\right]
=
\left[\begin{array}{c}
C' \\ C' \\ \vdots \\ C' \\ \hline C' \\ C' \\ \vdots \\ C'
\end{array}\right].
$$

Species Interaction Matrix

$$
\left[\begin{array}{cccc|cccc}
1 & 0 & \dots & 0 & f_{E_1,C_1} & f_{E_1,C_2} & \cdots & f_{E_1,C_H} \\
0 & 1 & \dots & 0 & f_{E_2,C_1} & f_{E_2,C_2} & \cdots & f_{E_2,C_H} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \dots & 1 & f_{E_K,C_1} & f_{E_K,C_2} & \cdots & f_{E_K,C_H} \\
\hline
f_{E_1,C_1} & f_{E_2,C_1} & \cdots & f_{E_K,C_1} & 1 & f_{C_1,C_2} & \cdots & f_{C_1,C_H} \\
f_{E_1,C_2} & f_{E_2,C_2} & \cdots & f_{E_K,C_2} & f_{C_1,C_2} & 1 & \cdots & f_{C_2,C_H} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
f_{E_1,C_H} & f_{E_2,C_H} & \cdots & f_{E_K,C_H} & f_{C_1,C_H}, & f_{C_2,C_H} & \cdots & 1
\end{array}\right]
\left[\begin{array}{c}
p_{E_1} \\ p_{E_2} \\ \vdots \\ p_{E_K} \\ \hline p_{C_1} \\ p_{C_2} \\ \vdots \\ p_{C_H}
\end{array}\right]
=
\left[\begin{array}{c}
1/K \\ 1/K \\ \vdots \\ 1/K \\ \hline 1/K \\ 1/K \\ \vdots \\ 1/K
\end{array}\right]
$$

Species Interaction Matrix

$$\left[\begin{array}{cccc|cccc}
1 & 0 & \cdots & 0 & f_{E_1,C_1} & f_{E_1,C_2} & \cdots & f_{E_1,C_H} \\
0 & 1 & \cdots & 0 & f_{E_2,C_1} & f_{E_2,C_2} & \cdots & f_{E_2,C_H} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 1 & f_{E_K,C_1} & f_{E_K,C_2} & \cdots & f_{E_K,C_H} \\
\hline
f_{E_1,C_1} & f_{E_2,C_1} & \cdots & f_{E_K,C_1} & 1 & f_{C_1,C_2} & \cdots & f_{C_1,C_H} \\
f_{E_1,C_2} & f_{E_2,C_2} & \cdots & f_{E_K,C_2} & f_{C_1,C_2} & 1 & \cdots & f_{C_2,C_H} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
f_{E_1,C_H} & f_{E_2,C_H} & \cdots & f_{E_K,C_H} & f_{C_1,C_H}, & f_{C_2,C_H} & \cdots & 1
\end{array}\right]
\left[\begin{array}{c}
K \cdot p_{E_1} \\
K \cdot p_{E_2} \\
\vdots \\
K \cdot p_{E_K} \\
\hline
K \cdot p_{C_1} \\
K \cdot p_{C_2} \\
\vdots \\
K \cdot p_{C_H}
\end{array}\right]
=
\left[\begin{array}{c}
1 \\ 1 \\ \vdots \\ 1 \\ \hline 1 \\ 1 \\ \vdots \\ 1
\end{array}\right]$$

XCSSS solution → MWPSLE solution

- Assume exact cover solution of K subsets.
- The vector $\boldsymbol{y}$ below solves the MWPSLE.

$$
\left[\begin{array}{cccc|cccc}
1 & 0 & \dots & 0 & f_{E_1,C_1} & f_{E_1,C_2} & \cdots & f_{E_1,C_H} \\
0 & 1 & \dots & 0 & f_{E_2,C_1} & f_{E_2,C_2} & \cdots & f_{E_2,C_H} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \dots & 1 & f_{E_K,C_1} & f_{E_K,C_2} & \cdots & f_{E_K,C_H} \\
\hline
f_{E_1,C_1} & f_{E_2,C_1} & \cdots & f_{E_K,C_1} & 1 & f_{C_1,C_2} & \cdots & f_{C_1,C_H} \\
f_{E_1,C_2} & f_{E_2,C_2} & \cdots & f_{E_K,C_2} & f_{C_1,C_2} & 1 & \cdots & f_{C_2,C_H} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
f_{E_1,C_H} & f_{E_2,C_H} & \cdots & f_{E_K,C_H} & f_{C_1,C_H}, & f_{C_2,C_H} & \cdots & 1
\end{array}\right]
\left[\begin{array}{c}
1 \\ 1 \\ \vdots \\ 1 \\ \hline 0 \\ 0 \\ \vdots \\ 0
\end{array}\right]
=
\left[\begin{array}{c}
1 \\ 1 \\ \vdots \\ 1 \\ \hline 1 \\ 1 \\ \vdots \\ 1
\end{array}\right]
$$

XCSSS solution ← MWPSLE solution

- Assume a non-negative vector $\boldsymbol{y}$ such that
  .
  $$M_{RFS} \cdot \boldsymbol{y} = \mathbf{1}$$
- Also assume that $\boldsymbol{y}$ has exactly K positive components.

$$\begin{bmatrix} 1 & a_{1,2} & \dots & a_{1,|C|} \\ a_{2,1} & 1 & \dots & a_{2,|C|} \\ \vdots & \vdots & \ddots & \vdots \\ a_{|C|,1} & a_{|C|,2} & \dots & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{|C|} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

- First we show all positive components of $\boldsymbol{y_i}$ must be 1:
  - Since all matrix coefficients $a_{ii} = 1$, and since all $\boldsymbol{y_i}$ and $\boldsymbol{a_{ij}}$ are non-negative, it follows that no $\boldsymbol{y_i}$ can exceed 1: $\forall i : y_i \leq 1$
  - Now since $\sum y_i = K$ then $\forall i : y_i = \{0, 1\}$ with exactly K components equal to one, and all others equal to zero.

XCSSS solution ← MWPSLE solution (continued)

$$\begin{bmatrix} 1 & a_{1,2} & \dots & a_{1,|C|} \\ a_{2,1} & 1 & \dots & a_{2,|C|} \\ \vdots & \vdots & \ddots & \vdots \\ a_{|C|,1} & a_{|C|,2} & \dots & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{|C|} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

- Next we show for any two distinct positive solution components y_i and y_j ($i \neq j$), the corresponding a_{ij} must be zero: $\forall i, j : (y_i = y_j = 1 \land i \neq j) \Rightarrow a_{i,j} = 0$
  - Proof by contradiction: If a_{ij} were not zero, then it would be positive and would add to the left hand side of both equations i and j . But since both y_i and y_j equal one, the left hand sides of equations i and j will then exceed one. Thus equations i and j will not be satisfied.

MENDEL 2014 results

Table 2: Raw Results from Runs on USA Today 200 Sudoku Puzzle Book

PUZZLE No. (within level)	DIFFICULTY LEVEL			PUZZLE No. (within level)	DIFFICULTY LEVEL		
	EASY	*MEDIUM*	*HARD*		*EASY*	*MEDIUM*	*HARD*
1	580	432	2622	35	291	1154	6187
2	146	2094	3596	36	125	1276	4131
3	90	365	2028	37	697	913	Not Solved
4	323	326	1987	38	98	382	831
5	254	317	2059	39	166	416	3225
6	521	706	Not Solved	40	335	508	817
7	558	728	685	41	372	379	580
8	342	1103	3341	42	93	749	1751
9	134	905	Not Solved	43	216	1369	4526
10	285	361	Not Solved	44	312	490	Not Solved
11	79	766	911	45	297	266	Not Solved
12	72	141	859	46	34	164	1161
13	58	229	2611	47	133	1028	248
14	245	2829	Not Solved	48	162	180	968
15	344	922	3640	49	232	1535	689
16	356	280	Not Solved	50	121	492	1360
17	140	425	2397	51	400	616	1077
18	447	696	1044	52	80	471	Not Solved
19	513	489	775	53	802	844	676
20	242	279	392	54	185	250	617
21	638	634	3630	55	343	479	9891
22	255	435	1066	56	90	361	Not Solved
23	314	125	826	57	206	900	1156

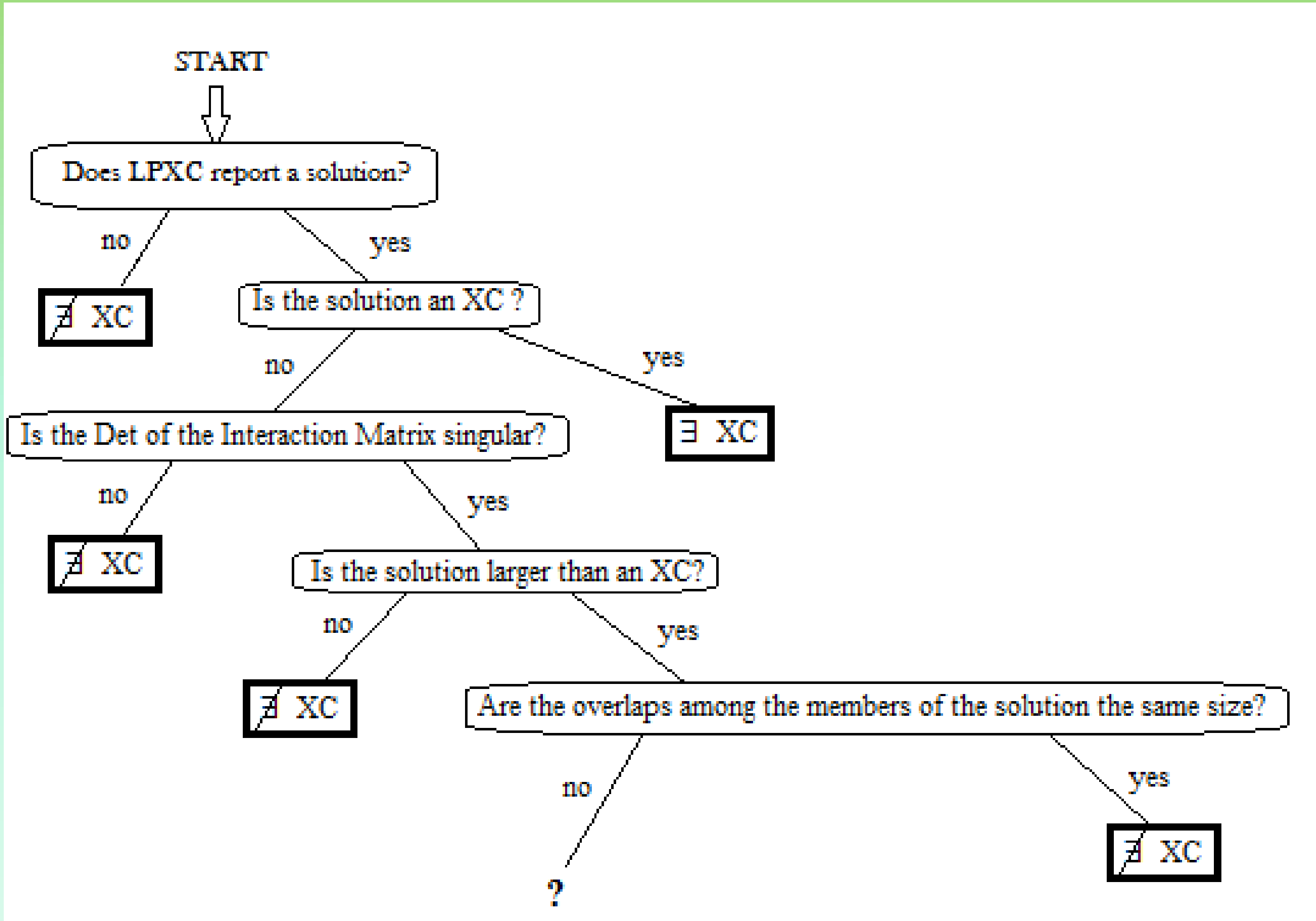
Table 2: Raw Results from Runs on USA Today 200 Sudoku Puzzle Book

PUZZLE No. (within level)	DIFFICULTY LEVEL			PUZZLE No. (within level)	DIFFICULTY LEVEL		
	EASY	*MEDIUM*	*HARD*		*EASY*	*MEDIUM*	*HARD*
1	580	432	2622	35	291	1154	6187
2	146	2094	3596	36	125	1276	4131
3	90	365	2028	37	697	913	Not Solved
4	323	326	1987	38	98	382	831
5	254	317	2059	39	166	416	3225
6	521	706	Not Solved	40	335	508	817
7	558	728	685	41	372	379	580
8	342	1103	3341	42	93	749	1751
9	134	905	Not Solved	43	216	1369	4526
10	285	361	Not Solved	44	312	490	Not Solved
11	79	766	911	45	297	266	Not Solved
12	72	141	859	46	34	164	1161
13	58	229	2611	47	133	1028	248
14	245	2829	Not Solved	48	162	180	968
15	344	922	3640	49	232	1535	689
16	356	280	Not Solved	50	121	492	1360
17	140	425	2397	51	400	616	1077
18	447	696	1044	52	80	471	Not Solved
19	513	489	775	53	802	844	676
20	242	279	392	54	185	250	617
21	638	634	3630	55	343	479	9891
22	255	435	1066	56	90	361	Not Solved
23	314	125	826	57	206	900	1156
24	269	326	1480	58	222	182	789
25	295	897	4555	59	319	612	2922
26	107	694	669	60	162	206	Not Solved
27	57	802	2211	61	468	740	1321
28	126	416	1025	62	80	876	4351
29	579	349	872	63	367	114	
30	141	855	Not Solved	64	89	368	
31	132	218	3317	65	1002	647	
32	128	2918	2356	66	96	205	
33	238	792	583	67	272	464	
34	511	275	1147	68	155	572	

Table 1: Summarized Results of Runs on USA Today 200 Sudoku Puzzle Book

Statistic	*EASY*	*MEDIUM*	*HARD*	**OVERALL**
# of puzzles solved total # of puzzles	68 68	68 68	50 62	186 198
solution rate (%)	100%	100%	81%	94%
Mean time to solution (generations) (standard deviation)	272.7 (193.3)	652.0 (534.5)	2039.2 (1778.6)	988.0 (1213.5)

Decision Process



RFSS-LP (Linear Programming)

Table 2: Raw Results from Runs on USA Today 200 Sudoku Puzzle Book

PUZZLE No. (within level)	DIFFICULTY LEVEL				PUZZLE No. (within level)	DIFFICULTY LEVEL			
	EASY	*MEDIUM*	*HARD*			*EASY*	*MEDIUM*	*HARD*	
1	580	432	2622	solved	35	291	1154	6187	
2	146	2094	3596	solved	36	125	1276	4131	
3	90	365	2028	solved	37	697	913	~~Not Solved~~	solved
4	323	326	1987	solved	38	98	382	831	
5	254	317	2059	solved	39	166	416	3225	
6	521	706	~~Not Solved~~	solved	40	335	508	817	
7	558	728	685	solved	41	372	379	580	
8	342	1103	3341	solved	42	93	749	1751	
9	134	905	Not Solved	~~solved~~	43	216	1369	4526	
10	285	361	Not Solved	~~solved~~	44	312	490	Not Solved	~~solved~~
11	79	766	911		45	297	266	Not Solved	~~solved~~
12	72	141	859		46	34	164	1161	
13	58	229	2611		47	133	1028	248	
14	245	2829	Not Solved	~~solved~~	48	162	180	968	
15	344	922	3640		49	232	1535	689	
16	356	280	~~Not Solved~~	solved	50	121	492	1360	
17	140	425	2397		51	400	616	1077	
18	447	696	1044		52	80	471	Not Solved	~~solved~~
19	513	489	775		53	802	844	676	
20	242	279	392		54	185	250	617	
21	638	634	3630		55	343	479	9891	
22	255	435	1066		56	90	361	Not Solved	~~solved~~
23	314	125	826		57	206	900	1156	
24	269	326	1480		58	222	182	789	
25	295	897	4555		59	319	612	2922	
26	107	694	669		60	162	206	~~Not Solved~~	solved
27	57	802	2211		61	468	740	1321	
28	126	416	1025		62	80	876	4351	
29	579	349	872		63	367	114		
30	141	855	Not Solved	~~solved~~	64	89	368		
31	132	218	3317		65	1002	647		
32	128	2918	2356		66	96	205		
33	238	792	583		67	272	464		
34	511	275	1147		68	155	572		

RFSS-LP (INTEGER Programming)

Table 2: Raw Results from Runs on USA Today 200 Sudoku Puzzle Book

PUZZLE No. (within level)	DIFFICULTY LEVEL				PUZZLE No. (within level)	DIFFICULTY LEVEL			
	EASY	*MEDIUM*	*HARD*			*EASY*	*MEDIUM*	*HARD*	
1	580	432	2622	solved	35	291	1154	6187	
2	146	2094	3596	solved	36	125	1276	4131	
3	90	365	2028	solved	37	697	913	~~Not Solved~~	solved
4	323	326	1987	solved	38	98	382	831	
5	254	317	2059	solved	39	166	416	3225	
6	521	706	~~Not Solved~~	solved	40	335	508	817	
7	558	728	685	solved	41	372	379	580	
8	342	1103	3341	solved	42	93	749	1751	
9	134	905	Not Solved	solved	43	216	1369	4526	
10	285	361	Not Solved	solved	44	312	490	Not Solved	solved
11	79	766	911		45	297	266	Not Solved	solved
12	72	141	859		46	34	164	1161	
13	58	229	2611		47	133	1028	248	
14	245	2829	Not Solved	solved	48	162	180	968	
15	344	922	3640		49	232	1535	689	
16	356	280	~~Not Solved~~	solved	50	121	492	1360	
17	140	425	2397		51	400	616	1077	
18	447	696	1044		52	80	471	Not Solved	solved
19	513	489	775		53	802	844	676	
20	242	279	392		54	185	250	617	
21	638	634	3630		55	343	479	9891	
22	255	435	1066		56	90	361	Not Solved	solved
23	314	125	826		57	206	900	1156	
24	269	326	1480		58	222	182	789	
25	295	897	4555		59	319	612	2922	
26	107	694	669		60	162	206	~~Not Solved~~	solved
27	57	802	2211		61	468	740	1321	
28	126	416	1025		62	80	876	4351	
29	579	349	872		63	367	114		
30	141	855	Not Solved	solved	64	89	368		
31	132	218	3317		65	1002	647		
32	128	2918	2356		66	96	205		
33	238	792	583		67	272	464		
34	511	275	1147		68	155	572		

Supersudoku

7						8	D					9	4		
D		3	9		7			5	C						
0						1				4		F		E	
				B	A			3		9		2		D	0
	E	2		3	8		A	0	D		5				1
			D	1							9				2
6					2	B				3		0	5	8	A
		7		C	5				1		E				
E			6			5									
		A	2				7	1					C		8
	0			4	1	9		B							
	8	C	4			2		A		5					6
	7		1		3	D					4			6	E
			3	6					B	2		8			
	5				E			9				A		F	
A		6	0		4			C						1	7

RFSS Solution

7						8	D					9	4		
D		3	9		7			5	C						
0						1				4		F		E	
				B	A			3		9		2		D	0
	E	2		3	8		A	0	D		5				1
			D	1							9				2
6					2	B				3		0	5	8	A
		7		C	5				1		E				
E			6			5									
		A	2				7	1					C		8
	0			4	1	9		B							
	8	C	4			2		A		5					6
	7		1		3	D					4			6	E
			3	6					B	2		8			
	5				E			9				A		F	
A		6	0		4			C						1	7

7	6	B	5	2	0	8	D	E	F	1	A	9	4	3	C
D	4	3	9	F	7	E	6	5	C	0	2	1	8	A	B
0	2	8	A	9	C	1	3	D	6	4	B	F	7	E	5
1	C	F	E	B	A	4	5	3	8	9	7	2	6	D	0
4	E	2	C	3	8	6	A	0	D	B	5	7	F	9	1
5	3	0	D	1	F	7	4	6	A	8	9	B	E	C	2
6	9	1	F	D	2	B	E	4	7	3	C	0	5	8	A
8	A	7	B	C	5	0	9	2	1	F	E	6	D	4	3
E	1	D	6	A	B	5	C	F	2	7	8	4	3	0	9
9	F	A	2	E	6	3	7	1	4	D	0	5	C	B	8
3	0	5	7	4	1	9	8	B	E	C	6	D	A	2	F
B	8	C	4	0	D	2	F	A	9	5	3	E	1	7	6
F	7	9	1	5	3	D	B	8	0	A	4	C	2	6	E
C	D	E	3	6	9	A	1	7	B	2	F	8	0	5	4
2	5	4	8	7	E	C	0	9	3	6	1	A	B	F	D
A	B	6	0	8	4	F	2	C	5	E	D	3	9	1	7

Hypersudoku

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

1. Only one numeral per cell
2. Exactly one of each numeral per row
3. Exactly one of each numeral per column
4. Exactly one of each numeral per region
5. **Exactly one of each numeral per shaded region**

Implications

- M_{RFS} is $|C|$ by $|C|$ in size , while M_{SM} is $|C|$ by $|X|$ (and $|X|$ can grow much exponentially in $|C|$, and vice versa)
- Are the linear approaches practical?
- Note that these NP-complete problems (e.g., X3C, MWPSLE, 0-1 Integer Programming) are solvable in polynomial time if the matrix (M_{RFS} or M_{SM}) is **non-singular**. (Can solve some singular matrices.)
- Perhaps M_{RFS} is non-singular when M_{SM} is, and vice versa.
- M_{RFS} works with arbitrary $|X|$ (XrC), even with continuous sets. Only requires set intersections.