Kakuro as a Constraint Problem

Helmut Simonis

Cork Constraint Computation Centre
Dept of Computer Science
University College Cork
Ireland

MODREF08

Outline

Introduction

- Discuss different models for solving Kakuro puzzles
- Finite domain constraints mimic human solving
- Nearly all published puzzles can be solved by a GAC *alldifferent-sum* without making guesses
- We can generate more difficult puzzles
- Very strong competition from (even naive) SAT

Related Work

- Sudoku as a Constraint Problem (Simonis, 2005)
- Solitaire Battleship (Smith, 2006)
- Teaching constraints through logic puzzles (Szeredi, 2003)
- Crossword Puzzles (Van Hentenryck 89, LeProvost&Wallace 92)

Some Definitions

Definition

A puzzle is called *valid* if it has solutions.

Definition

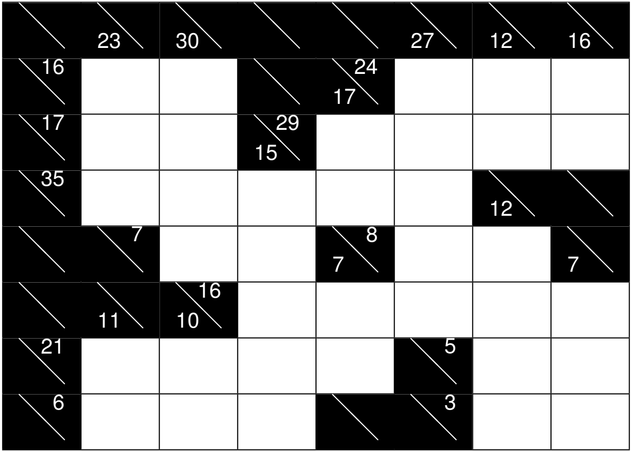
A puzzle is called *well posed* if it has exactly one solution.

Definition

A well posed puzzle is *(locally) minimal*, if no hint can be removed without admitting additional solutions.

Outline

Problem



Rules

1. Rectilinear grid of black and white cells.
2. Black cells may contain hints (integer numbers) for cells to the right/below.
3. Fill white cells with numbers 1..9 such that:
    1. The sum of a continuous block of white cell in horizontal (or vertical) direction must be equal to the hint given in the black cell to the left (above).
    2. All numbers in a continuous block of white cells must be pairwise different.
4. Do not guess values, all values should be deduced.

Problem Solution

\	23\	30\	\	\	27\	12\	16\
\16	9	7	\	17\24	8	7	9
\17	8	9	15\29	8	9	5	7
\35	6	8	5	9	7	12\	\
\	\7	6	1	7\8	2	6	7\
\	11\	10\16	4	6	1	3	2
\21	8	9	3	1	\5	1	4
\6	3	1	2	\	\3	2	1

Outline

Finite Domain Model

Grid G: all white cells
Hints H: set of hints $< I, v >$
Variables x_i the value stored in cell i

$$\forall_{i \in G} : x_i \in [1, 9]$$

$$\forall_{<I,v> \in H} : \texttt{alldifferent}(\{x_i | i \in I\})$$

$$\forall_{<I,v> \in H} : \sum_{i \in I} x_i = v$$

MIP/SAT

0/1 variables $y_{i,j}$ indicate if cell i contains value j

$$\forall_{i \in G}, \forall_{j \in [1,9]} : y_{ij} \in \{0,1\}$$

$$\forall_{i \in G} : \sum_{j \in [1,9]} y_{ij} = 1$$

$$\forall_{<I,v> \in H}, \forall_{j \in [1,9]} : \sum_{i \in I} y_{ij} \leq 1$$

$$\forall_{<I,v> \in H} : \sum_{i \in I} \sum_{j \in [1,9]} j * y_{ij} = v$$

Data Sources

- **big** A single very large (124x90) puzzle instance from Nikoli.
- **mix** Puzzles from the Penpa Mix puzzle collections of Nikoli.
- **giants** Another puzzle collection from Nikoli, containing large scale examples (up to 32x46).
- **kakuro** A special issue from Nikoli on Kakuro.
- **jnp** A collection of number puzzles which contains some Kakuro instances.
- **suzuki** Another collection of different puzzle types containing a number of Kakuro puzzles.

FD Details

- Search
  - Complete, *input-order*/*indomain* variable/value selection
- Shaving
  - One iteration to achieve singleton arc-consistency before search
  - Test each value for each variable to see if inconsistent
  - Can remove many infeasible values before search is started
  - Considered ”not-fair” by many human puzzle solvers

FD + shaving (ECLiPSe 5.10)

Set	X	Y	K	Setup	Shave	Total	Time		Back	
							Avg	Max	Avg	Max
big	124	90	1	0.00	0.00	100.00	258.09	258.09	577710.00	577710
giants	32	22	46	0.00	54.35	89.13	10.01	287.37	26776.51	760019
giants	32	42	6	0.00	33.33	83.33	6.25	15.53	13705.80	36702
giants	32	46	1	0.00	0.00	100.00	68.92	68.92	161256.00	161256
jnp	9	9	8	0.00	100.00	100.00	0.01	0.01	0.00	0
jnp	10	10	8	0.00	87.50	100.00	0.03	0.07	3.88	31
jnp	12	12	8	0.00	50.00	100.00	0.04	0.08	4.75	32
kakuro	10	14	39	0.00	100.00	100.00	0.02	0.04	0.00	0
kakuro	16	16	44	0.00	81.82	100.00	0.13	2.26	150.34	6210
kakuro	32	22	18	0.00	61.11	100.00	1.03	8.82	2413.11	31867
mix	12	12	12	0.00	100.00	100.00	0.01	0.02	0.00	0
mix	16	16	70	0.00	87.14	100.00	0.08	0.58	31.11	1615
mix	32	22	8	0.00	50.00	100.00	0.85	2.14	1019.75	3565
suzuki	20	12	44	0.00	97.73	100.00	0.04	0.12	0.43	19
All			313	0.00	80.51	98.08	2.63	287.37	6403.28	760019

MIP (ECLiPSe + CLP/CBC Coin-OR)

Group	X	Y	K	Nr Vars	Solved	Avg Time	Max Time
big	124	90	1	n/a	n/a	n/a	n/a
giants	32	22	46	4465.17	10.87	98.81	241.31
giants	32	42	6	8742.00	0.00	n/a	n/a
giants	32	46	1	9423.00	0.00	n/a	n/a
jnp	9	9	8	366.75	100.00	0.85	4.65
jnp	10	10	8	572.62	100.00	3.41	10.40
jnp	12	12	8	820.12	100.00	9.54	31.18
kakuro	10	14	39	745.38	100.00	1.02	6.90
kakuro	16	16	44	1487.45	95.45	29.81	151.25
kakuro	32	22	18	4351.00	27.78	153.23	301.24
mix	12	12	12	744.00	100.00	1.10	6.25
mix	16	16	70	1494.00	92.86	23.03	248.53
mix	32	22	8	4341.38	12.50	3.94	3.94
suzuki	20	12	44	1357.77	100.00	17.00	287.30
All			313	2122.33	75.71	20.78	301.24

SAT (Minisat+)

Set	X	Y	K	Solved	Restart	Conflict	Decisions		Time	
							Avg	Max	Avg	Max
big	124	90	1	0.00	n/a	n/a	n/a	n/a	n/a	n/a
giants	32	22	46	100.00	15.13	91793.39	292493.80	817837	51.64	173.06
giants	32	42	6	100.00	17.00	165484.33	695334.00	920827	178.34	254.92
giants	32	46	1	100.00	17.00	163183.00	778007.00	778007	204.01	204.01
jnp	9	9	8	100.00	5.50	1437.75	4282.75	7412	0.43	0.57
jnp	10	10	8	100.00	8.38	5198.00	14062.75	28060	1.34	2.10
jnp	12	12	8	100.00	10.38	11475.12	29130.00	48752	2.75	4.91
kakuro	10	14	39	100.00	7.59	3939.18	11064.38	23566	1.30	2.38
kakuro	16	16	44	100.00	11.43	18389.02	52642.30	135559	5.24	13.06
kakuro	32	22	18	100.00	14.56	61504.61	224880.67	340253	41.65	74.16
mix	12	12	12	100.00	7.50	3714.08	10850.58	21390	1.11	1.80
mix	16	16	70	100.00	10.97	15185.57	46195.66	101888	5.15	11.06
mix	32	22	8	100.00	15.25	79101.00	264553.25	365393	40.65	67.36
suzuki	20	12	44	100.00	9.73	9082.39	30027.75	53245	3.60	7.58
All			313	99.68	11.15	31196.45	103787.51	920827	17.87	254.92

Observations

- FD: No instance is solved by propagation alone
- Even with shaving, only 80 percent are solved
- (Naive) SAT model solves more instances than FD within timeout
- (Naive) MIP not competitive

Outline

Example after Initial Propagation

	23	30			27	12	16
16	 789	 789		24 17	... ..6 789	... ..6 789	 789
17	... ... .89	... ... .89	29 15	... ... .89	... .56 789	..3 456 ...	 789
35	... .56 78.	... ..6 789	123 456 789	... ... .89	123 456 789	12	
	7	..3 456 ...	123 4.. ...	8 7	123 456 7..	123 456 7..	7
	11	16 10	123 456 789	123 456 ...	123 456 789	123 456 789	123 45. ...
21	 789	 789	123 456 ...	123 456 ...	5	123 4.. ...	123 4.. ...
6	.23 4.. ...	123	123		3	12.	12. ... ...

Problem: Missing Propagation

```
[X1,X2,X3,X4,X5] :: 1..9,
alldifferent([X1,X2,X3,X4,X5]),
X1+X2+X3+X4+X5 #= 15
```

No domain reduction

$$\sum_{k \in I} x_k = N$$

$$\overline{x_i} = N - \sum_{k \in I, k \neq i} \underline{x_k}$$

$$\underline{x_i} = N - \sum_{k \in I, k \neq i} \overline{x_k}$$

Problem: Missing Propagation

```
[X1,X2,X3,X4,X5] :: 1..9,
alldifferent([X1,X2,X3,X4,X5]),
X1+X2+X3+X4+X5 #= 15
```

No domain reduction

$$\sum_{k \in I} x_k = 15$$

$$\overline{x_i} = 15 - \sum_{k \in I, k \neq i} 1 = 11$$

$$\underline{x_i} = 15 - \sum_{k \in I, k \neq i} 9 = -21$$

Problem: Missing Propagation

```
[X1,X2,X3,X4,X5] :: 1..9,
alldifferent([X1,X2,X3,X4,X5]),
X1+X2+X3+X4+X5 #= 15
```

Possible domain reduction

$$\sum_{k \in I} x_k = 15$$

$$\overline{x_i} = 15 - (1 + 2 + 3 + 4) = 5$$

$$\underline{x_i} = 15 - (5 + 4 + 3 + 2) = 1$$

First Idea: Value Removal

- For a given number of variables and given sum, only some values can be used
- All other values can be removed from domains
- Easy to precompute a priori
- Imposed as a domain restriction

Example after Value Removal

	23	30			27	12	16
16	9	... ... 7.9		24 17	 789	 789	... ... 7.9
17	... ... .89	... ... .89	29 15	... ... .89	... .5. 789	... .5. 789	... ... 7.9
35	... ..6 .89	... ..6 789	5	... ... .89	... .56 789	12	
	7	6	123 45. ...	8 7	123 .56 7..	123 .56 ...	7
	11	16 10	123 4.. ...	123 4.6 ...	123 4.6 ...	123 4.6 ...	12. 4.. ...
21	.23 456 789	123 4.6 789	123 45. ...	123 456 ...	5	123 4.. ...	12. 4.. ...
6	.23	123	123		3	12.	12. ... ...

Example after Value Removal + Propagation

	23	30			27	12	16
16	9	7		24 17	8	7	9
17	8	9	29 15	8	9	5	7
35	6	8	5	9	7	12	
	7	6	1	8 7	2	6	7
	11	16 10	.23 4.. ...	... 4.6 ...	1	.23 4.. ...	.2. 4.. ...
21	... ... .89	... ... .89	.23 4.. ...	123	5	123	.2. 4.. ...
6	.23	12.	.23		3	12.	12. ... ...

FD + Removed values

							Time		Back	
Set	X	Y	K	Setup	Shave	Total	Avg	Max	Avg	Max
big	124	90	1	0.00	100.00	100.00	1.76	1.76	0.00	0
giants	32	22	46	4.35	100.00	100.00	0.08	0.18	0.00	0
giants	32	42	6	0.00	100.00	100.00	0.17	0.22	0.00	0
giants	32	46	1	0.00	100.00	100.00	0.18	0.18	0.00	0
jnp	9	9	8	50.00	100.00	100.00	0.01	0.02	0.00	0
jnp	10	10	8	25.00	100.00	100.00	0.01	0.04	0.00	0
jnp	12	12	8	0.00	100.00	100.00	0.01	0.02	0.00	0
kakuro	10	14	39	76.92	100.00	100.00	0.01	0.04	0.00	0
kakuro	16	16	44	27.27	100.00	100.00	0.03	0.05	0.00	0
kakuro	32	22	18	5.56	100.00	100.00	0.07	0.11	0.00	0
mix	12	12	12	91.67	100.00	100.00	0.01	0.02	0.00	0
mix	16	16	70	27.14	100.00	100.00	0.02	0.08	0.00	0
mix	32	22	8	12.50	100.00	100.00	0.07	0.13	0.00	0
suzuki	20	12	44	2.27	100.00	100.00	0.02	0.05	0.00	0
All			313	26.52	100.00	100.00	0.04	1.76	0.00	0

SAT + Removed values

Set	X	Y	K	Solved	Restart	Conflict	Decisions		Time	
							Avg	Max	Avg	Max
big	124	90	1	100.00	16.00	87910.00	1006558.00	1006558	434.58	434.58
giants	32	22	46	100.00	4.65	2022.70	8786.11	50232	1.80	7.61
giants	32	42	6	100.00	7.00	3434.17	25951.83	51882	6.40	11.37
giants	32	46	1	100.00	7.00	2734.00	25118.00	25118	6.04	6.04
jnp	9	9	8	100.00	1.12	57.00	180.25	413	0.06	0.08
jnp	10	10	8	100.00	1.88	205.88	643.12	2908	0.12	0.35
jnp	12	12	8	100.00	2.25	290.25	1068.62	4031	0.18	0.42
kakuro	10	14	39	100.00	1.05	11.74	44.23	526	0.06	0.12
kakuro	16	16	44	100.00	1.82	194.64	745.39	4935	0.22	0.85
kakuro	32	22	18	100.00	3.28	554.50	2941.78	9599	0.86	1.84
mix	12	12	12	100.00	1.00	6.83	35.00	136	0.06	0.08
mix	16	16	70	100.00	1.53	134.59	570.49	6116	0.20	0.80
mix	32	22	8	100.00	3.12	638.25	3146.00	12087	0.94	2.52
suzuki	20	12	44	100.00	2.32	315.55	1091.64	7616	0.23	0.93
All			313	100.00	2.39	818.58	5775.44	1006558	1.99	434.58

Arc-consistent alldifferent-sum

- There is still missed propagation
- This is caused by decomposition into *sum* and *alldifferent*
- We need arc-consistent *alldifferent-sum*
- Every value for variable in constraint is supported
  - No such constraint in global constraint catalog
  - Write new global constraint? No...
  - Simulate by other global constraint
  - Use general mechanism

GCC with Cost

- Model proposed by M. Carlsson
- Version of *GCC* (Global cardinality constraint)
- How often values can occur in set of variables
- Cost matrix defines cost of assigning value to variable
- Can simulate *alldifferent-sum*
- Simulation not arc consistent

Propia

- Generalized Propagation (Le Provost & Wallace, 92)
- Deduce domain restrictions from all possible solutions
- Arc consistent for explicit representation
- In that case equivalent to *table* constraint

Counting Valid Tuples

	Arity							
Value	2	3	4	5	6	7	8	9
3	2	-	-	-	-	-	-	-
4	2	-	-	-	-	-	-	-
5	4	-	-	-	-	-	-	-
6	4	6	-	-	-	-	-	-
7	6	6	-	-	-	-	-	-
8	6	12	-	-	-	-	-	-
9	8	18	-	-	-	-	-	-
10	8	24	24	-	-	-	-	-
11	8	30	24	-	-	-	-	-
12	6	42	48	-	-	-	-	-
13	6	42	72	-	-	-	-	-
14	4	48	120	-	-	-	-	-
15	4	48	144	120	-	-	-	-
16	2	48	192	120	-	-	-	-
17	2	42	216	240	-	-	-	-
18	-	42	264	360	-	-	-	-
19	-	30	264	600	-	-	-	-
20	-	24	288	720	-	-	-	-
21	-	18	264	960	720	-	-	-
22	-	12	264	1080	720	-	-	-
23	-	6	216	1320	1440	-	-	-
24	-	6	192	1320	2160	-	-	-

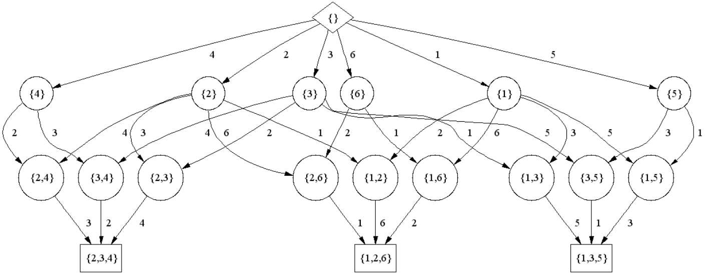
Counting Valid Tuples (2)

	Arity							
Value	2	3	4	5	6	7	8	9
25	-	-	144	1440	2880	-	-	-
26	-	-	120	1320	3600	-	-	-
27	-	-	72	1320	5040	-	-	-
28	-	-	48	1080	5040	5040	-	-
29	-	-	24	960	5760	5040	-	-
30	-	-	24	720	5760	10080	-	-
31	-	-	-	600	5760	10080	-	-
32	-	-	-	360	5040	15120	-	-
33	-	-	-	240	5040	15120	-	-
34	-	-	-	120	3600	20160	-	-
35	-	-	-	120	2880	20160	-	-
36	-	-	-	-	2160	20160	40320	-
37	-	-	-	-	1440	15120	40320	-
38	-	-	-	-	720	15120	40320	-
39	-	-	-	-	720	10080	40320	-
40	-	-	-	-	-	10080	40320	-
41	-	-	-	-	-	5040	40320	-
42	-	-	-	-	-	5040	40320	-
43	-	-	-	-	-	-	40320	-
44	-	-	-	-	-	-	40320	-
45	-	-	-	-	-	-	-	362880

More Compact Representation

- Arity 8 and 9 can be handled by *alldifferent* + domain reduction
- Tuple Compression (Katsirelos & Walsh, 2007)
  - Does not work for permutations
- *Case* Constraint (Beldiceanu & Carlsson)
- *Regular* Constraint (Pesant, 2004)
  - Accepts tuples which fit regular expression
  - Arc-consistent propagation
  - Just state disjunction of valid tuples
  - Generate compact automaton
  - Gecode model provided by C. Schulte

Automaton (3 variables with sum 9)



Is this enough?

- For most instances, arc-consistent *alldifferent-sum* + propagation is enough
- Three instances found which require more propagation, only one seems intentional
- Much more common for Sudoku puzzles
  - Simple extension can solve these three instances without shaving

(Current) Best ECLiPSe Variant

FD+Reorder+Propia+Removal+Redundant Model+Shaving

							Time		Back	
Set	X	Y	K	Setup	Shave	Total	Avg	Max	Avg	Max
big	124	90	1	100.00	100.00	100.00	2.73	2.73	0.00	0
giants	32	22	46	100.00	100.00	100.00	0.17	1.28	0.00	0
giants	32	42	6	100.00	100.00	100.00	0.27	0.35	0.00	0
giants	32	46	1	100.00	100.00	100.00	0.28	0.28	0.00	0
jnp	9	9	8	100.00	100.00	100.00	0.01	0.01	0.00	0
jnp	10	10	8	100.00	100.00	100.00	0.01	0.03	0.00	0
jnp	12	12	8	87.50	100.00	100.00	0.03	0.06	0.00	0
kakuro	10	14	39	100.00	100.00	100.00	0.02	0.05	0.00	0
kakuro	16	16	44	100.00	100.00	100.00	0.04	0.07	0.00	0
kakuro	32	22	18	94.44	100.00	100.00	0.11	0.22	0.00	0
mix	12	12	12	100.00	100.00	100.00	0.02	0.05	0.00	0
mix	16	16	70	100.00	100.00	100.00	0.04	0.09	0.00	0
mix	32	22	8	100.00	100.00	100.00	0.09	0.12	0.00	0
suzuki	20	12	44	97.73	100.00	100.00	0.04	0.09	0.00	0
All			313	99.04	100.00	100.00	0.07	2.73	0.00	0

Comparison: Gecode

						Time		Ratio Gecode/ECLiPSe		
Set	X	Y	K	Setup	Total	Avg	Max	Min	Avg	Max
big	124	90	1	100.00	100.00	0.64	0.64	0.23	0.23	0.23
giants	32	22	46	100.00	100.00	0.19	0.89	0.10	1.18	4.27
giants	32	42	6	100.00	100.00	0.19	0.44	0.24	0.71	1.63
giants	32	46	1	100.00	100.00	0.40	0.40	1.38	1.38	1.38
jnp	9	9	8	100.00	100.00	0.00	0.01	0.50	0.83	1.00
jnp	10	10	8	100.00	100.00	0.03	0.08	0.67	2.36	4.00
jnp	12	12	8	87.50	100.00	0.09	0.48	0.50	4.29	24.00
kakuro	10	14	39	100.00	100.00	0.00	0.03	0.50	0.68	1.50
kakuro	16	16	44	100.00	100.00	0.08	0.47	0.13	1.97	9.33
kakuro	32	22	18	94.44	100.00	0.13	0.57	0.20	1.29	6.33
mix	12	12	12	100.00	100.00	0.00	0.01	0.50	0.83	1.00
mix	16	16	70	100.00	100.00	0.06	0.49	0.14	1.73	12.25
mix	32	22	8	100.00	100.00	0.14	0.36	0.60	1.34	3.60
suzuki	20	12	44	97.73	100.00	0.08	0.43	0.25	2.01	8.67
All			313	99.04	100.00	0.08	0.89	0.10	1.67	24.00

Example: Not solved by GAC alldifferent-sum

3	9	2	6	1	45	
1	26 17	8	9	3	6	42
5	8	9	6	10 8	3	7
2	3	27 11	3	7	... ... .89	... ... .89
21 9	6	5	2	1	4	3
5	7 12	6	1	14 11	... .5. .8.	... ..6 ..9
4	7	12	22	9	... .5. 7..	... ..6 .8.
3 9	2	1	7 24	2	1	4
1	3	2	7	7 10	2	5
6	29	5	... ... .89	... ... 78.	 789	
2	15	4	... ... .89	.23		

Outline

Idea

- Perform Initial domain reduction
- Setup basic, decomposed model
- Add GAC *alldifferent-sum* for increasing arity
- Natural for humans, start with simpler reasoning
- Some puzzles require full GAC for all sizes

Grading of Instances

Set	Grade	X	Y	K	Setup	P2	P3	P4	P5	P6	P7	P8	P9
big	hard	124	90	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
giants	hard	32	22	46	4.35	4.35	6.52	43.48	54.35	65.22	82.61	86.96	100.00
giants	hard	32	42	6	0.00	0.00	0.00	0.00	33.33	83.33	83.33	83.33	100.00
giants	hard	32	46	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
jnp	easy	9	9	8	50.00	50.00	62.50	100.00	100.00	100.00	100.00	100.00	100.00
jnp	medium	10	10	8	25.00	25.00	37.50	62.50	75.00	100.00	100.00	100.00	100.00
jnp	medium	12	12	8	0.00	0.00	0.00	25.00	50.00	75.00	87.50	87.50	87.50
kakuro	easy	10	14	24	91.67	91.67	100.00	100.00	100.00	100.00	100.00	100.00	100.00
kakuro	medium	10	14	12	66.67	66.67	91.67	91.67	100.00	100.00	100.00	100.00	100.00
kakuro	hard	10	14	3	0.00	0.00	33.33	100.00	100.00	100.00	100.00	100.00	100.00
kakuro	easy	16	16	5	80.00	80.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
kakuro	medium	16	16	22	27.27	27.27	54.55	72.73	100.00	100.00	100.00	100.00	100.00
kakuro	hard	16	16	17	11.76	11.76	11.76	29.41	58.82	76.47	94.12	94.12	100.00
kakuro	hard	32	22	18	5.56	5.56	11.11	44.44	77.78	88.89	88.89	94.44	94.44
mix	easy	12	12	8	87.50	87.50	100.00	100.00	100.00	100.00	100.00	100.00	100.00
mix	medium	12	12	4	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
mix	medium	16	16	30	33.33	33.33	46.67	70.00	96.67	96.67	100.00	100.00	100.00
mix	hard	16	16	40	22.50	22.50	32.50	77.50	85.00	92.50	97.50	97.50	100.00
mix	hard	32	22	8	12.50	12.50	12.50	37.50	62.50	100.00	100.00	100.00	100.00
suzuki	easy	20	12	15	0.00	0.00	13.33	60.00	100.00	100.00	100.00	100.00	100.00
suzuki	medium	20	12	20	5.00	5.00	5.00	20.00	70.00	95.00	100.00	100.00	100.00
suzuki	hard	20	12	9	0.00	0.00	0.00	11.11	11.11	33.33	66.67	66.67	88.89
All				313	26.52	26.52	35.46	60.06	78.27	87.86	93.93	94.89	99.04

Are the published puzzles minimal?

- Can we remove hints while keeping unique solution?
- *Complete* Removal: Remove hint, no restriction on variables
- *Partial* Removal: Remove hint, keep *alldifferent* on variables
- We can, indeed, remove many hints (even with greedy selection)
- This makes the puzzle more difficult again

Reduction Summary

Set	X	Y	K	Complete Min	Complete Avg	Complete Max	Partial Min	Partial Avg	Partial Max
giants	32	22	10	5.51	6.86	8.40	11.42	14.26	20.44
jnp	9	9	8	6.25	11.02	16.67	6.25	14.44	17.86
jnp	10	10	8	5.56	10.38	15.79	8.33	12.10	15.79
jnp	12	12	8	7.41	10.18	16.07	10.00	13.39	18.52
kakuro	10	14	39	3.85	10.48	15.52	12.07	21.64	36.21
kakuro	16	16	44	3.41	8.48	12.50	9.76	17.68	25.00
mix	12	12	12	4.17	9.85	14.29	13.46	21.06	37.50
mix	16	16	70	4.26	9.35	13.21	11.36	17.58	26.19
suzuki	20	12	44	6.38	10.39	13.46	11.96	18.22	26.53
All			243	3.41	9.60	16.67	6.25	17.98	37.50

Partial Reduction Results

Set	X	Y	K	Setup	Shave	Total	Time Avg	Time Max	Back Avg	Back Max
giants	32	22	10	0.00	60.00	100.00	15.98	49.85	4.40	34
jnp	9	9	8	62.50	100.00	100.00	0.02	0.06	0.00	0
jnp	10	10	8	50.00	100.00	100.00	0.03	0.07	0.00	0
jnp	12	12	8	12.50	100.00	100.00	0.16	0.48	0.00	0
kakuro	10	14	39	41.03	100.00	100.00	0.04	0.18	0.00	0
kakuro	16	16	44	13.64	90.91	100.00	0.47	6.37	1.80	65
mix	12	12	12	41.67	100.00	100.00	0.02	0.04	0.00	0
mix	16	16	70	21.43	98.57	100.00	0.94	37.02	0.00	0
suzuki	20	12	44	22.73	100.00	100.00	0.12	0.52	0.00	0
All			243	25.51	96.30	100.00	1.05	49.85	0.51	65

Conclusions

- Naive SAT more stable than naive CP on examples shown
  - No single explanation for this
  - Better (packaged) search needed
  - Unexploited constraint interaction
- With care, high level FD implementations can compete
- (Published) Kakuro puzzles not as interesting as Sudoku
  - GAC *alldifferent-sum* is enough
  - Generating puzzles is much harder than for Sudoku, though
  - We can generate much harder Kakuro puzzles