CROWDSOURCING INSIGHTS INTO PROBLEM STRUCTURE FOR SCIENTIFIC DISCOVERY

CORNELL UNIVERSITY

Intelligent Information Systems Institute

Bart Selman

Carla P. Gomes

Ronan Le Bras (now @ AI2)

Stefano Ermon (now @ Stanford)

Motivation: Automated Scientific Discovery

Can we use modern automated reasoning tools for scientific discovery?

Science and scientific discovery are arguably among the most impressive demonstrations of human intelligence.

We will consider two areas:

1) Finite Mathematics

2) Materials Science

I Discovery in Finite Mathematics --- A Brief History

In the early days of computing, there was significant optimism with the discovery of first-order theorem proving procedures.

But, the raw size of search spaces for interesting math proved to be a daunting challenge. In a sense, the representation language (first-order logic) is too general and unconstrained to effectively explore.

Still, some success stories of automated mathematical discoveries, involving "finite mathematics." (Objects of interest are finite.)

We only consider cases of true discovery of previously unknown results (i.e. no "re-discovery").

Discoveries in Discrete Math

A) *1976* --- Four Color Thm. Four colors suffices to color a planar map. (Haken and Appel) Problem was open since ***1852***. Essentially solved using an exhaustive "proof by cases." (two thousand special sub-maps were analyzed. (90% human / 10% machine)

Proof is now accepted and has been replicated. Considered somewhat unsatisfying / "inelegant" from a math perspective, because the proof does not provide further insights into why the thm is true. Also, how do we know the computer proof is correct? (Do we?)

B) *1996* --- Robbins algebras are Boolean algebras. Concerns sets of axioms. (McCune) Open since ***1933***. Proof strategy: clever syntactic re-write of axioms. Several months of cpu time to find the re-write steps. "Semi-human" readable. (30% human / 70% machine)

C) *2014* --- Erdos Discrepancy Problem. Property of sequences of +1/-1. E.g. **+1 +1 -1 +1 -1 -1 +1 -1 …**

Does the sum (absolute value) over subsequences of this sequence stay within a certain bound, C?

Erdos conjectured **(*1932*)**: If sequence gets long enough "discrepancy" exceeds any value of C. Lisitsa and Konev (2014) showed every sequence of **1161 or longer** exceeds bound of **2** (i.e. C = 2). *But, remarkably, they also found a +1/-1 sequence of **1160** elements that stayed within bound of 2.*

(10% human / 90% machine)

Obtained with Boolean **Satisfiability** solver (20 hrs). Proof trace of **13 gigabytes**. Longest proof in history. Proof **verified** by proof checker. (Contrast two previous results.)

Each result represents a clear new mathematical advance. But, unfortunately, approaches do not provide much insight into the underlying problem domain.

For example, can we understand how certain complex objects (e.g. 1160 long +1/-1 sequence or coloring of sub-map) can be constructed directly using a human-understandable strategy?

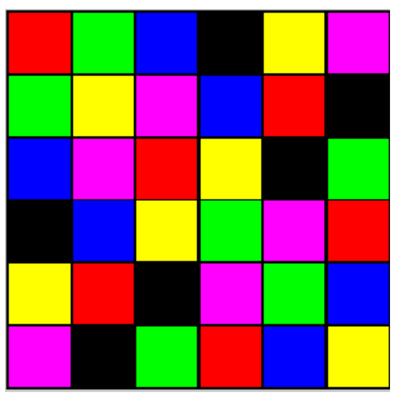
Example Domain:
The *Spatially-balanced Latin square (SBLS)* problem

Problem Definition:

An ***SBLS*** of order *n* is an *n* x *n* square grid in which:

- **Each symbol appears exactly once in each row and column (*Latin square* structure; multiplication table of quasi-group*)*.**

SBLS of order 6



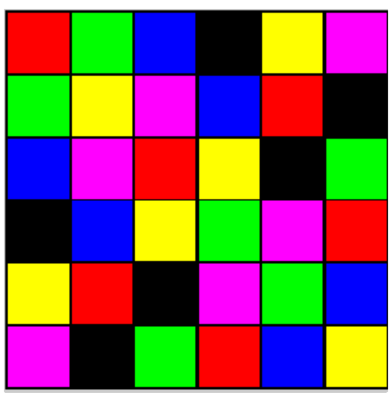
Our Challenge Domain:
The *Spatially-Balanced Latin Square (SBLS)* problem

Problem Definition:

An ***SBLS*** of order *n* is an *n* x *n* square grid in which:

- **Each symbol appears exactly once in each row and column (*Latin square* structure; multiplication table of quasi-group*)*.**
- **The average distance (column-wise) of a pair of symbols is the same for any pair *(Balanced structure)*.**

SBLS of order 6



Row Distance for pairs of colors

1	2	3	4	5
4	1	1	3	2
3	2	2	1	1
2	4	5	3	1
3	4	4	1	2
1	1	1	2	3

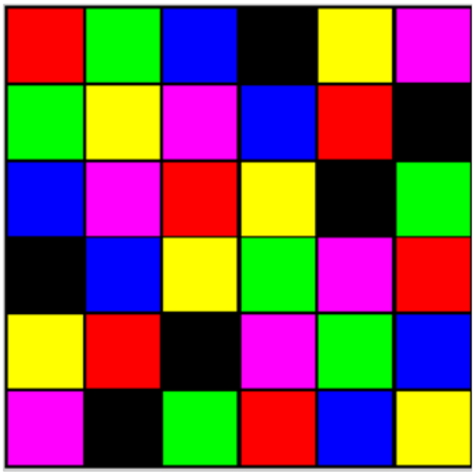
Total Row Distance (pair)

Average Row Distance (pair)

The *Spatially-Balanced Latin Square (SBLS)* problem

SBLS of order 6

Row Distance for pairs of colors



	1	2	3	4	5
	4	1	1	3	2
	3	2	2	1	1
	2	4	5	3	1
	3	4	1	1	2
	1	1	2	2	3
Total Row Distance (pair)	14	14	14	14	14
Average Row Distance (pair)	2.33	2.33	2.33	2.33	2.33

14 (avg. 2.33) for all pairs. Use in “experimental design.”

In Computational Sustainability: to limit use of fertilizers.

Designs used by farmers in New York State!

Finding SBLS is very hard.

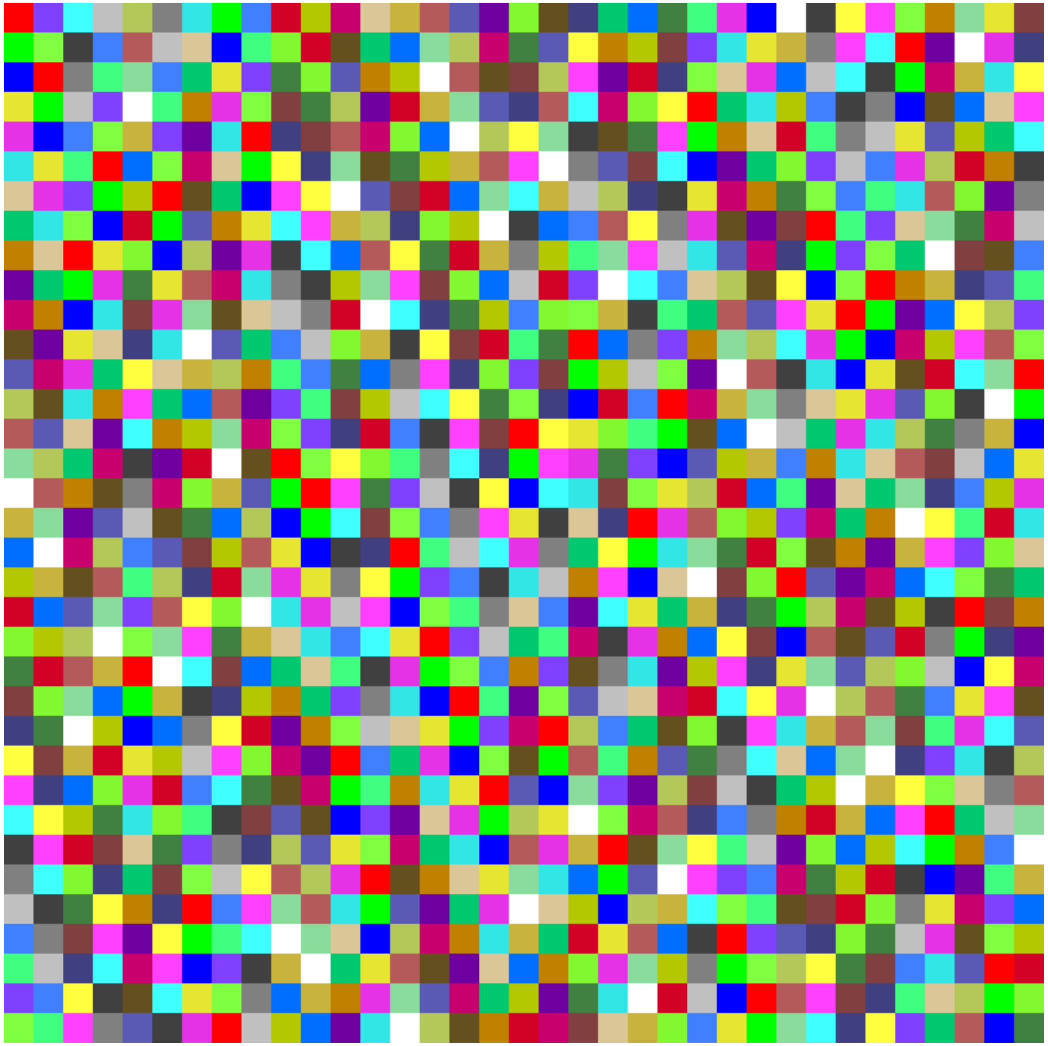
Approach	Order	Time (s)	Reference
Constraint Programming (CP)	9	241	[Gomes and Sellmann, CP'04]
IDWalk *(metaheuristic)*	9	4.5	[Neveu et al., CP'04]
Self-symmetry-based Streamlined CP	14	5,434	[Gomes and Sellmann, CP'04]
Composition-based Streamlined CP	18	107K	[Gomes and Sellmann, CP'04]
Streamlined Local Search	**35**	**1.2M**	[Smith et al., IJCAI'05]

Approach: use lots of problem specific structure to guide search.

Earlier known designs, only up to order 6.

The balancing condition (of each pair!) makes the problem *very* hard.

Largest ever found 35x35 --- 330 hrs cpu time.



This work: 50x50 in << 1 ms

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	49	47	45	43	41	39	37	35	33	31	29	27	25	23	21	19	17	15	13	11	9	7	5	3	1
3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	50	47	44	41	38	35	32	29	26	23	20	17	14	11	8	5	2	1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46	49
4	8	12	16	20	24	28	32	36	40	44	48	49	45	41	37	33	29	25	21	17	13	9	5	1	3	7	11	15	19	23	27	31	35	39	43	47	50	46	42	38	34	30	26	22	18	14	10	6	2
5	10	15	20	25	30	35	40	45	50	46	41	36	31	26	21	16	11	6	1	4	9	14	19	24	29	34	39	44	49	47	42	37	32	27	22	17	12	7	2	3	8	13	18	23	28	33	38	43	48
6	12	18	24	30	36	42	48	47	41	35	29	23	17	11	5	1	7	13	19	25	31	37	43	49	46	40	34	28	22	16	10	4	2	8	14	20	26	32	38	44	50	45	39	33	27	21	15	9	3
7	14	21	28	35	42	49	45	38	31	24	17	10	3	4	11	18	25	32	39	46	48	41	34	27	20	13	6	1	8	15	22	29	36	43	50	44	37	30	23	16	9	2	5	12	19	26	33	40	47
8	16	24	32	40	48	45	37	29	21	13	5	3	11	19	27	35	43	50	42	34	26	18	10	2	6	14	22	30	38	46	47	39	31	23	15	7	1	9	17	25	33	41	49	44	36	28	20	12	4
9	18	27	36	45	47	38	29	20	11	2	7	16	25	34	43	49	40	31	22	13	4	5	14	23	32	41	50	42	33	24	15	6	3	12	21	30	39	48	44	35	26	17	8	1	10	19	28	37	46
10	20	30	40	50	41	31	21	11	1	9	19	29	39	49	42	32	22	12	2	8	18	28	38	48	43	33	23	13	3	7	17	27	37	47	44	34	24	14	4	6	16	26	36	46	45	35	25	15	5
11	22	33	44	46	35	24	13	2	9	20	31	42	48	37	26	15	4	7	18	29	40	50	39	28	17	6	5	16	27	38	49	41	30	19	8	3	14	25	36	47	43	32	21	10	1	12	23	34	45
12	24	36	48	41	29	17	5	7	19	31	43	46	34	22	10	2	14	26	38	50	39	27	15	3	9	21	33	45	44	32	20	8	4	16	28	40	49	37	25	13	1	11	23	35	47	42	30	18	6
13	26	39	49	36	23	10	3	16	29	42	46	33	20	7	6	19	32	45	43	30	17	4	9	22	35	48	40	27	14	1	12	25	38	50	37	24	11	2	15	28	41	47	34	21	8	5	18	31	44
14	28	42	45	31	17	3	11	25	39	48	34	20	6	8	22	36	50	37	23	9	5	19	33	47	40	26	12	2	16	30	44	43	29	15	1	13	27	41	46	32	18	4	10	24	38	49	35	21	7
15	30	45	41	26	11	4	19	34	49	37	22	7	8	23	38	48	33	18	3	12	27	42	44	29	14	1	16	31	46	40	25	10	5	20	35	50	36	21	6	9	24	39	47	32	17	2	13	28	43
16	32	48	37	21	5	11	27	43	42	26	10	6	22	38	47	31	15	1	17	33	49	36	20	4	12	28	44	41	25	9	7	23	39	46	30	14	2	18	34	50	35	19	3	13	29	45	40	24	8
17	34	50	33	16	1	18	35	49	32	15	2	19	36	48	31	14	3	20	37	47	30	13	4	21	38	46	29	12	5	22	39	45	28	11	6	23	40	44	27	10	7	24	41	43	26	9	8	25	42
18	36	47	29	11	7	25	43	40	22	4	14	32	50	33	15	3	21	39	44	26	8	10	28	46	37	19	1	17	35	48	30	12	6	24	42	41	23	5	13	31	49	34	16	2	20	38	45	27	9
19	38	44	25	6	13	32	50	31	12	7	26	45	37	18	1	20	39	43	24	5	14	33	49	30	11	8	27	46	36	17	2	21	40	42	23	4	15	34	48	29	10	9	28	47	35	16	3	22	41
20	40	41	21	1	19	39	42	22	2	18	38	43	23	3	17	37	44	24	4	16	36	45	25	5	15	35	46	26	6	14	34	47	27	7	13	33	48	28	8	12	32	49	29	9	11	31	50	30	10
21	42	38	17	4	25	46	34	13	8	29	50	30	9	12	33	47	26	5	16	37	43	22	1	20	41	39	18	3	24	45	35	14	7	28	49	31	10	11	32	48	27	6	15	36	44	23	2	19	40
22	44	35	13	9	31	48	26	4	18	40	39	17	5	27	49	30	8	14	36	43	21	1	23	45	34	12	10	32	47	25	3	19	41	38	16	6	28	50	29	7	15	37	42	20	2	24	46	33	11
23	46	32	9	14	37	41	18	5	28	50	27	4	19	42	36	13	10	33	45	22	1	24	47	31	8	15	38	40	17	6	29	49	26	3	20	43	35	12	11	34	44	21	2	25	48	30	7	16	39
24	48	29	5	19	43	34	10	14	38	39	15	9	33	44	20	4	28	49	25	1	23	47	30	6	18	42	35	11	13	37	40	16	8	32	45	21	3	27	50	26	2	22	46	31	7	17	41	36	12
25	50	26	1	24	49	27	2	23	48	28	3	22	47	29	4	21	46	30	5	20	45	31	6	19	44	32	7	18	43	33	8	17	42	34	9	16	41	35	10	15	40	36	11	14	39	37	12	13	38
26	49	23	3	29	46	20	6	32	43	17	9	35	40	14	12	38	37	11	15	41	34	8	18	44	31	5	21	47	28	2	24	50	25	1	27	48	22	4	30	45	19	7	33	42	16	10	36	39	13
27	47	20	7	34	40	13	14	41	33	6	21	48	26	1	28	46	19	8	35	39	12	15	42	32	5	22	49	25	2	29	45	18	9	36	38	11	16	43	31	4	23	50	24	3	30	44	17	10	37
28	45	17	11	39	34	6	22	50	23	5	33	40	12	16	44	29	1	27	46	18	10	38	35	7	21	49	24	4	32	41	13	15	43	30	2	26	47	19	9	37	36	8	20	48	25	3	31	42	14
29	43	14	15	44	28	1	30	42	13	16	45	27	2	31	41	12	17	46	26	3	32	40	11	18	47	25	4	33	39	10	19	48	24	5	34	38	9	20	49	23	6	35	37	8	21	50	22	7	36
30	41	11	19	49	22	8	38	33	3	27	44	14	16	46	25	5	35	36	6	24	47	17	13	43	28	2	32	39	9	21	50	20	10	40	31	1	29	42	12	18	48	23	7	37	34	4	26	45	15
31	39	8	23	47	16	15	46	24	7	38	32	1	30	40	9	22	48	17	14	45	25	6	37	33	2	29	41	10	21	49	18	13	44	26	5	36	34	3	28	42	11	20	50	19	12	43	27	4	35
32	37	5	27	42	10	22	47	15	17	49	20	12	44	25	7	39	30	2	34	35	3	29	40	8	24	45	13	19	50	18	14	46	23	9	41	28	4	36	33	1	31	38	6	26	43	11	21	48	16
33	35	2	31	37	4	29	39	6	27	41	8	25	43	10	23	45	12	21	47	14	19	49	16	17	50	18	15	48	20	13	46	22	11	44	24	9	42	26	7	40	28	5	38	30	3	36	32	1	34
34	33	1	35	32	2	36	31	3	37	30	4	38	29	5	39	28	6	40	27	7	41	26	8	42	25	9	43	24	10	44	23	11	45	22	12	46	21	13	47	20	14	48	19	15	49	18	16	50	17
35	31	4	39	27	8	43	23	12	47	19	16	50	15	20	46	11	24	42	7	28	38	3	32	34	1	36	30	5	40	26	9	44	22	13	48	18	17	49	14	21	45	10	25	41	6	29	37	2	33
36	29	7	43	22	14	50	15	21	44	8	28	37	1	35	30	6	42	23	13	49	16	20	45	9	27	38	2	34	31	5	41	24	12	48	17	19	46	10	26	39	3	33	32	4	40	25	11	47	18
37	27	10	47	17	20	44	7	30	34	3	40	24	13	50	14	23	41	4	33	31	6	43	21	16	48	11	26	38	1	36	28	9	46	18	19	45	8	29	35	2	39	25	12	49	15	22	42	5	32
38	25	13	50	12	26	37	1	39	24	14	49	11	27	36	2	40	23	15	48	10	28	35	3	41	22	16	47	9	29	34	4	42	21	17	46	8	30	33	5	43	20	18	45	7	31	32	6	44	19
39	23	16	46	7	32	30	9	48	14	25	37	2	41	21	18	44	5	34	28	11	50	12	27	35	4	43	19	20	42	3	36	26	13	49	10	29	33	6	45	17	22	40	1	38	24	15	47	8	31
40	21	19	42	2	38	23	17	44	4	36	25	15	46	6	34	27	13	48	8	32	29	11	50	10	30	31	9	49	12	28	33	7	47	14	26	35	5	45	16	24	37	3	43	18	22	39	1	41	20
41	19	22	38	3	44	16	25	35	6	47	13	28	32	9	50	10	31	29	12	48	7	34	26	15	45	4	37	23	18	42	1	40	20	21	39	2	43	17	24	36	5	46	14	27	33	8	49	11	30
42	17	25	34	8	50	9	33	26	16	43	1	41	18	24	35	7	49	10	32	27	15	44	2	40	19	23	36	6	48	11	31	28	14	45	3	39	20	22	37	5	47	12	30	29	13	46	4	38	21
43	15	28	30	13	45	2	41	17	26	32	11	47	4	39	19	24	34	9	49	6	37	21	22	36	7	50	8	35	23	20	38	5	48	10	33	25	18	40	3	46	12	31	27	16	42	1	44	14	29
44	13	31	26	18	39	5	49	8	36	21	23	34	10	47	3	41	16	28	29	15	42	2	46	11	33	24	20	37	7	50	6	38	19	25	32	12	45	1	43	14	30	27	17	40	4	48	9	35	22
45	11	34	22	23	33	12	44	1	46	10	35	21	24	32	13	43	2	47	9	36	20	25	31	14	42	3	48	8	37	19	26	30	15	41	4	49	7	38	18	27	29	16	40	5	50	6	39	17	28
46	9	37	18	28	27	19	36	10	45	1	47	8	38	17	29	26	20	35	11	44	2	48	7	39	16	30	25	21	34	12	43	3	49	6	40	15	31	24	22	33	13	42	4	50	5	41	14	32	23
47	7	40	14	33	21	26	28	19	35	12	42	5	49	2	45	9	38	16	31	23	24	30	17	37	10	44	3	50	4	43	11	36	18	29	25	22	32	15	39	8	46	1	48	6	41	13	34	20	27
48	5	43	10	38	15	33	20	28	25	23	30	18	35	13	40	8	45	3	50	2	46	7	41	12	36	17	31	22	26	27	21	32	16	37	11	42	6	47	1	49	4	44	9	39	14	34	19	29	24
49	3	46	6	43	9	40	12	37	15	34	18	31	21	28	24	25	27	22	30	19	33	16	36	13	39	10	42	7	45	4	48	1	50	2	47	5	44	8	41	11	38	14	35	17	32	20	29	23	26
50	1	49	2	48	3	47	4	46	5	45	6	44	7	43	8	42	9	41	10	40	11	39	12	38	13	37	14	36	15	35	16	34	17	33	18	32	19	31	20	30	21	29	22	28	23	27	24	26	25

How?

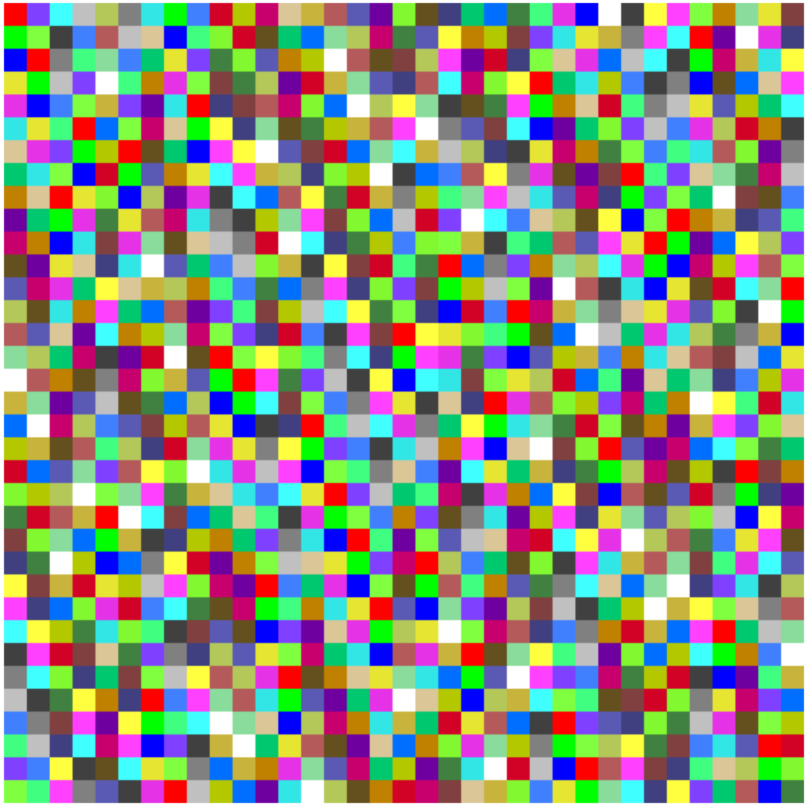
(Nothing special about 50x50. We can do 10Kx10K in a few seconds.)

(here)

Gedanken Experiment:

Given that a constructive algorithm (given any size n) exists, how would you find it? Hwk…

(Can use SAT solver.)



Outline

- Motivation
- Example Domain
- Proposed Framework
  - Overview of Streamlined Search
  - Taking advantage of Human Insights
  - Formal Description and Overview
  - Human-guided Streamlined Search
- Application to the *Spatially-balanced Latin square* problem
- Application to the *Weak Schur Number* problem
- Conclusions and Future work

Outline

- Motivation
- Example Domain
- Proposed Framework
  - Overview of Streamlined Search
  - Taking advantage of Human Insights
  - Formal Description and Overview
  - GUI for Human-guided Streamlined Search
- Application to the *Spatially-balanced Latin square* problem
- Application to the *Weak Schur Number* problem
- Conclusions and Future work

Proposed Framework:
Overview of Streamlined Combinatorial Search

Goal:

Exploit the **structure of some solutions** to **boost** the effectiveness of the **propagation mechanisms.**

Underlying Observation:

When one insists on maintaining the **full solution set**, there is a **hard practical limit** on the effectiveness of **constraint propagation** methods. Often, there is **no compact representation** for all the solutions.

Underlying Conjecture:

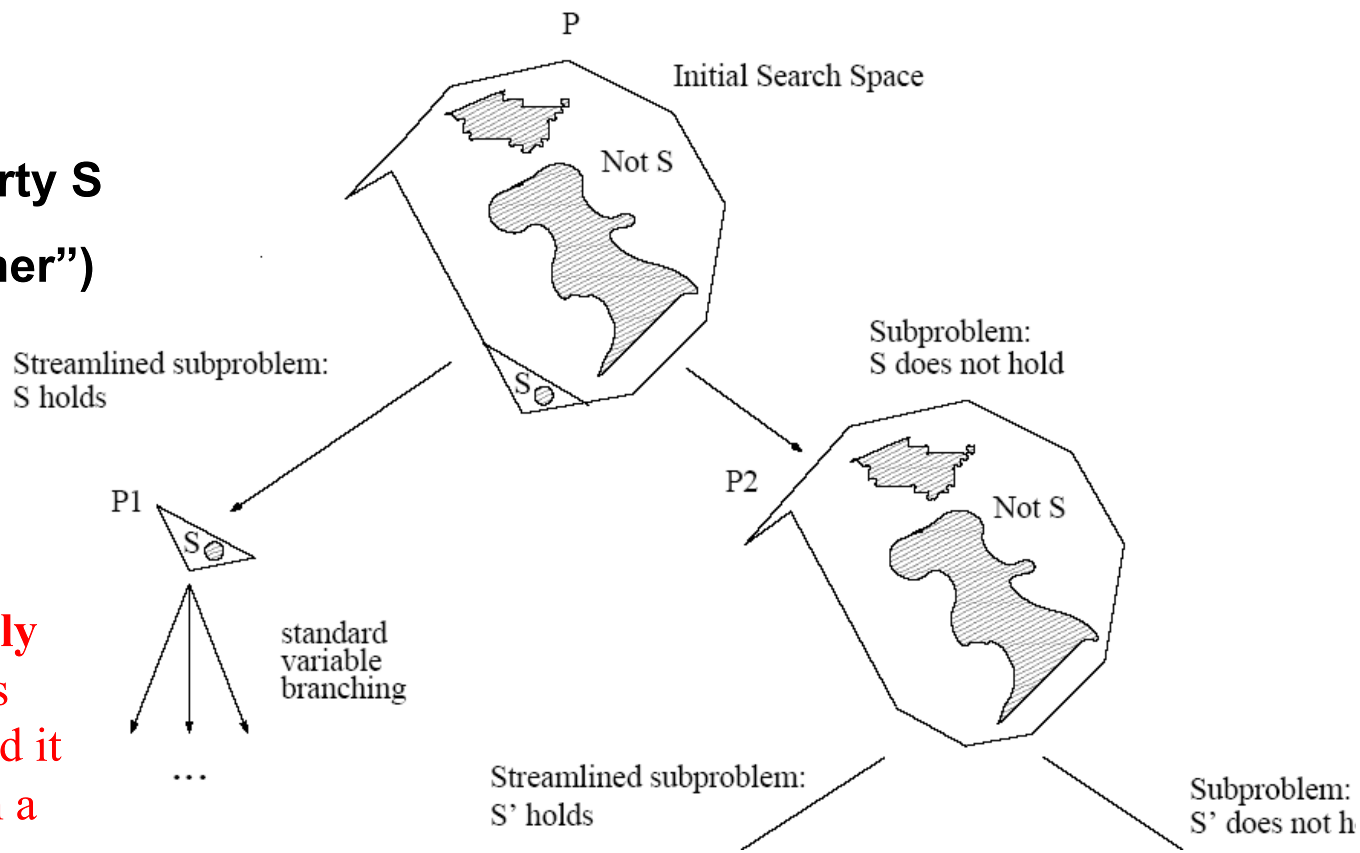
For many intricate **combinatorial problems** – if **solutions exist** – there will often be (highly) **regular ones**. **Can we find such solutions?**

Overview of Streamlined Search

Assert

global property S

(a “streamliner”)



P_1 is **substantially smaller** than its complement P_2 and it will benefit from a **stronger filtering** thereafter**.**

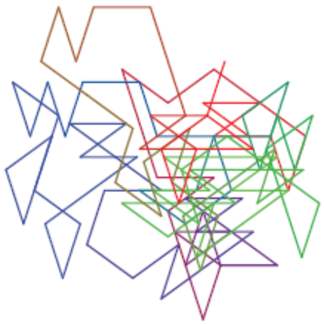
<u>Streamlined Search:</u>

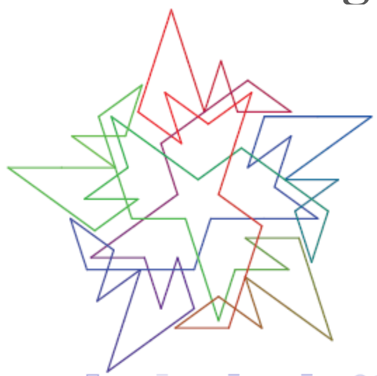
Strong **branching mechanisms** (by adding constraints based on **structure properties**) at **high levels** of the search tree.

[Gomes and Sellmann, CP’04]

Taking advantage of Human Insights

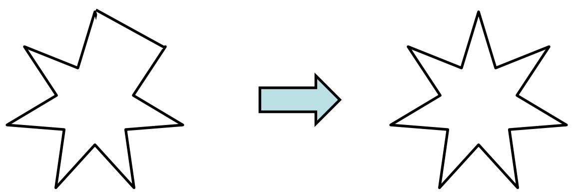
Recognizing Patterns and Regularities:





[Source: Marijn J.H. Heule, 2009, in work on van der Waerden numbers]

Correcting Irregularities:



Generalizing / Formalizing Regularities:

1	2	3
3	1	2
2	3	1

Cyclic Latin square of order 3

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

Cyclic Latin square of order 4

Proposed Framework:
Formal Description and Overview

$\mathcal{O} \leftarrow \emptyset$; // Conjectured streamliners
$\Gamma \leftarrow \emptyset$; // Search streamliners
$\rho \leftarrow \rho_0$; // Search parameter
$\mathcal{S} \leftarrow \emptyset$; // Solutions found
$\tau \leftarrow \mathit{false}$; // Timeout flag
repeat
  $\mathit{Solve}(P_\rho, \Gamma, t) \rightarrow (\mathcal{S}', \tau)$; // Search for new solutions
  if $\mathcal{S}' \cap \mathcal{S} \neq \emptyset$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}'$; // Case 1: successful search
    $\mathit{Analyze}(\mathcal{S}) \rightarrow \mathcal{O}'$; // Conjecture new streamliners
    $\mathcal{O} \leftarrow \mathcal{O} \cup \mathcal{O}'$;
    $\rho \leftarrow \rho + 1$;
  else if τ *is true* **then**
    Select $\Gamma' \subseteq \mathcal{O}$; // Case 2: timed-out failed search
    $\Gamma \leftarrow \Gamma \cup \Gamma'$; // Strengthen streamliners
  else
    Select $\Gamma' \subseteq \Gamma$; // Case 3: exhaustive failed search
    $\Gamma \leftarrow \Gamma \setminus \Gamma'$; // Weaken streamliners
    $\rho = max\{\rho : \mathcal{S}(\Gamma) \cap \mathcal{S}(P_\rho) \neq \emptyset\}+1$;
    Select $\Gamma'' \subseteq \Gamma'$; // Find next parameter of interest
    $\mathcal{O} \leftarrow \mathcal{O} \setminus \Gamma''$; // Drop unpromising streamliners
until $\mathcal{O} = \emptyset$;

Algorithm : `Discover-Construction` procedure for a given problem P, with parameter set ρ and timeout t.

Overview Approach --- Mimics Alg Discovery

```
O ← ∅;                                          // Conjectured streamliners
Γ ← ∅;                                             // Search streamliners
ρ ← ρ0;                                            // Search parameter
S ← ∅;                                              // Solutions found
τ ← false;                                            // Timeout flag
repeat
    Solve(P_ρ, Γ, t) → (S′, τ);   // Search for new solutions
    if S′ ∩ S ≠ ∅ then
        S ← S ∪ S′;                      // Case 1: successful search
        Analyze(S) → O′;   // Conjecture new streamliners
        O ← O ∪ O′;
        ρ ← ρ + 1;
    else if τ is true then
        Select Γ′ ⊆ O;        // Case 2: timed-out failed search
        Γ ← Γ ∪ Γ′;                          // Strengthen streamliners
    else
        Select Γ′ ⊆ Γ;      // Case 3: exhaustive failed search
        Γ ← Γ \ Γ′;                               // Weaken streamliners
        ρ = max{ρ : S(Γ) ∩ S(P_ρ) ≠ ∅}+1;
        Select Γ″ ⊆ Γ′;     // Find next parameter of interest
        O ← O \ Γ″;       // Drop unpromising streamliners
until O = ∅ ;
```

Algorithm : `Discover-Construction` procedure for a given problem P, with parameter set ρ and timeout t.

1. **Analyze** smaller size solutions, and **conjecture potential regularities** in the solutions. (Human insight.)
2. Validate through **streamlining** the observed regularities.
3. If the streamlined search **does not give a larger size solution**, the proposed regularity is quite likely **accidental** and one looks for a new pattern in the small scale solutions.
4. Otherwise, one proceeds by generating a number of **new solutions** that all contain the proposed **structural regularity** and are used to expand the solution set and to **reveal new regularities**.

Proposed Framework for Human-guided Streamlined Search

Overview on the *SBLS* problem

Search Parameters

n=*3*

Γ={}

Conjectured Streamliners

Start with first order of interest (n=3) and no streamliners (Γ={})

Solution Set

Overview on the *SBLS* problem

Search Parameters

n=3

Γ={}

Conjectured Streamliners

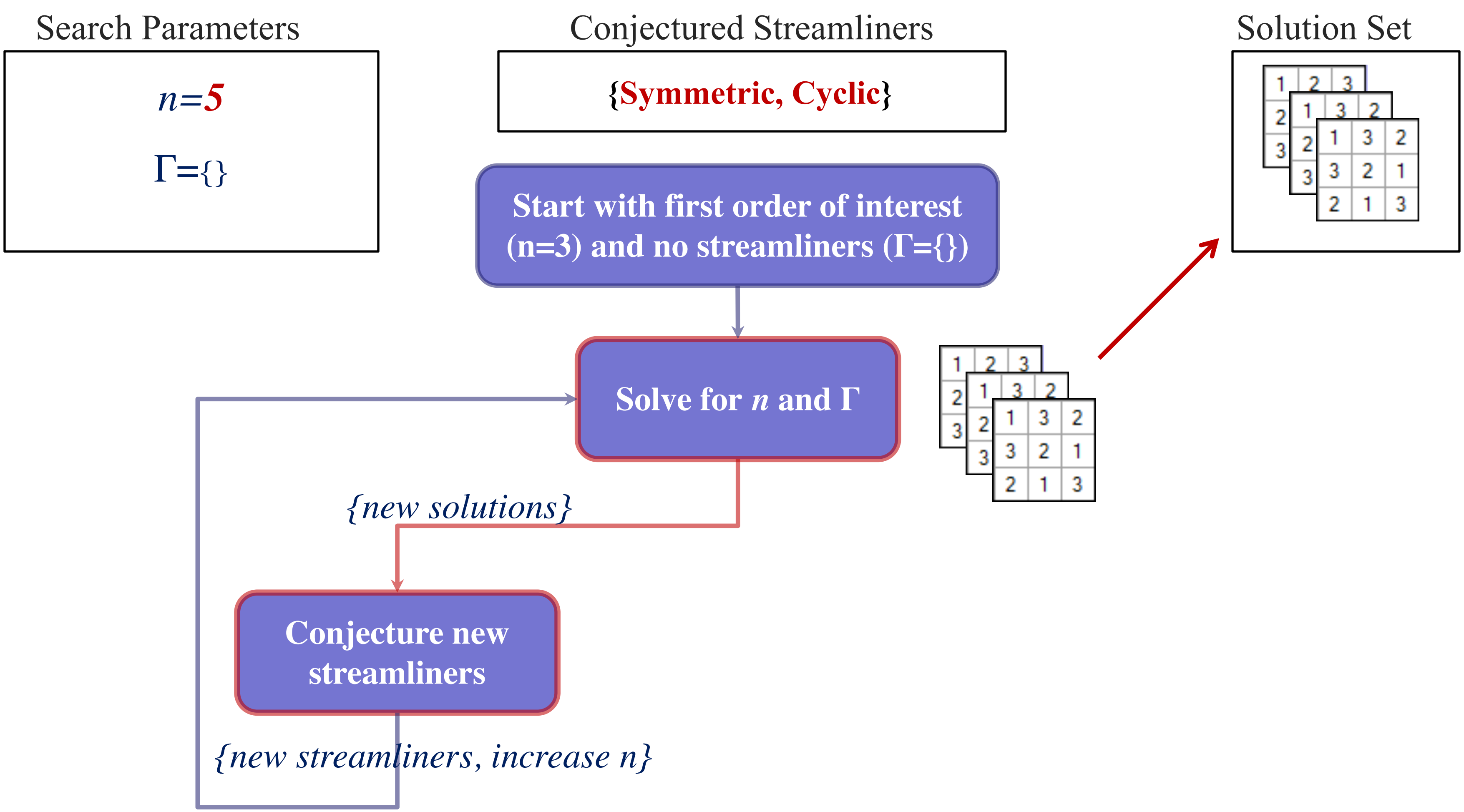
Solution Set

Start with lowest order of interest (n=3) and no streamliners (Γ={})

Solve for *n* and Γ

Proposed Framework for Human-guided Streamlined Search

Overview on the *SBLS* problem



Overview on the *SBLS* problem

Search Parameters

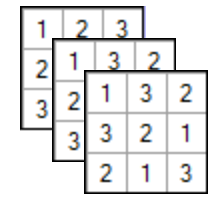
n=5

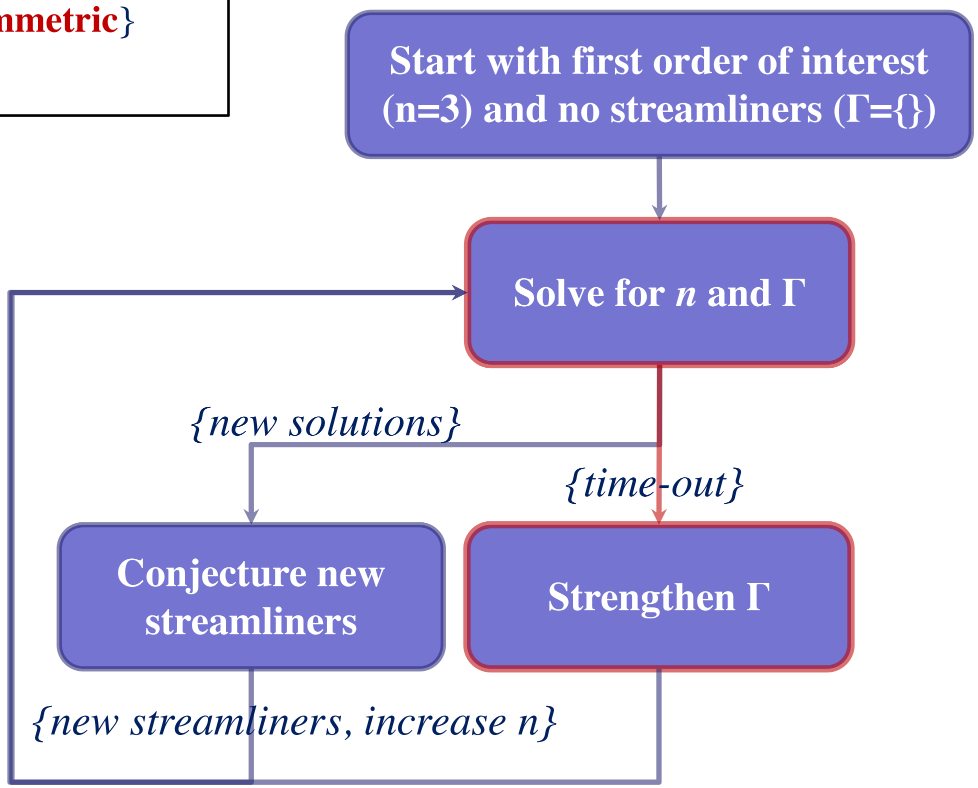
Γ={**Symmetric**}

Conjectured Streamliners

{Symmetric, Cyclic}

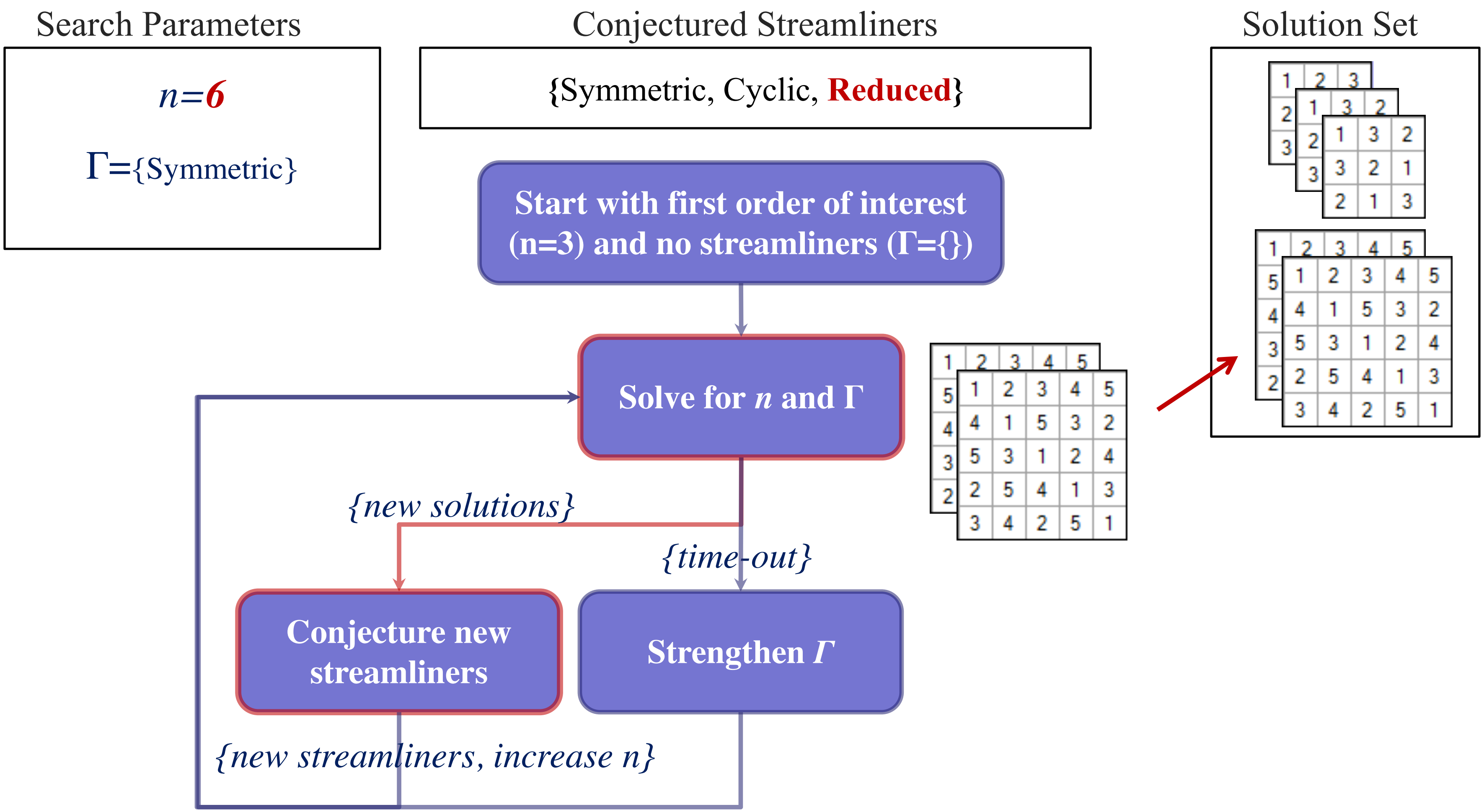
Solution Set





Proposed Framework for Human-guided Streamlined Search

Overview on the *SBLS* problem



CORNELL UNIVERSITY

Overview on the *SBLS* problem

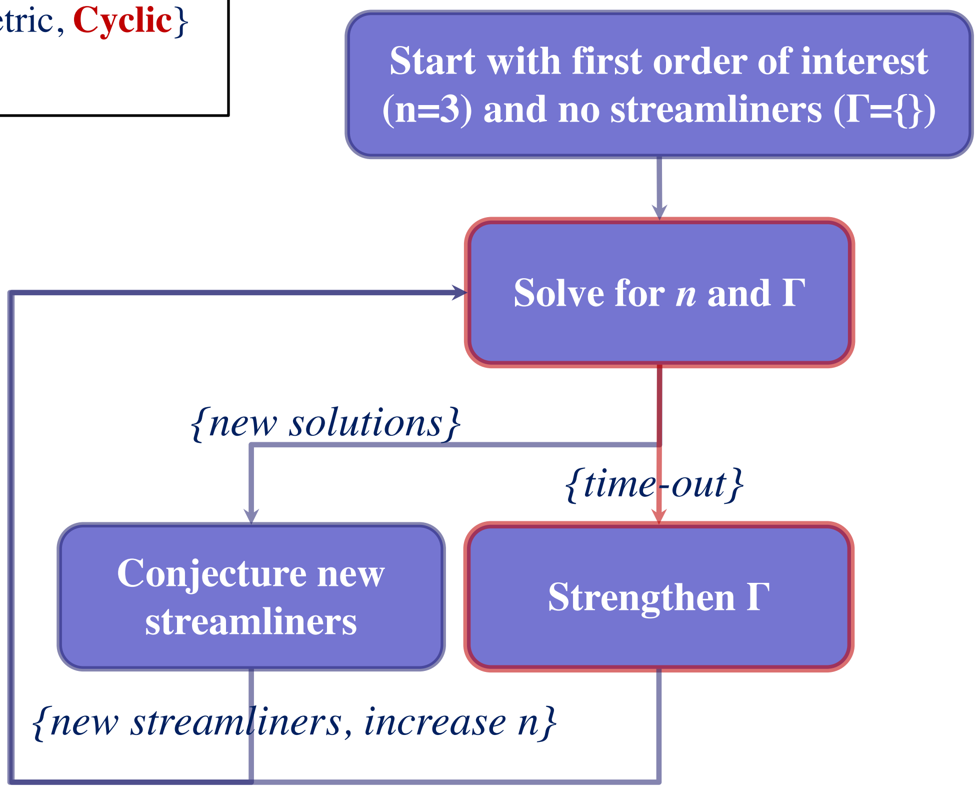
Search Parameters

n=6

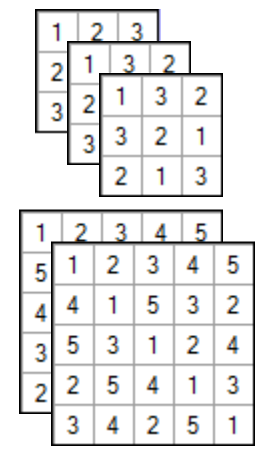
Γ={Symmetric, **Cyclic**}

Conjectured Streamliners

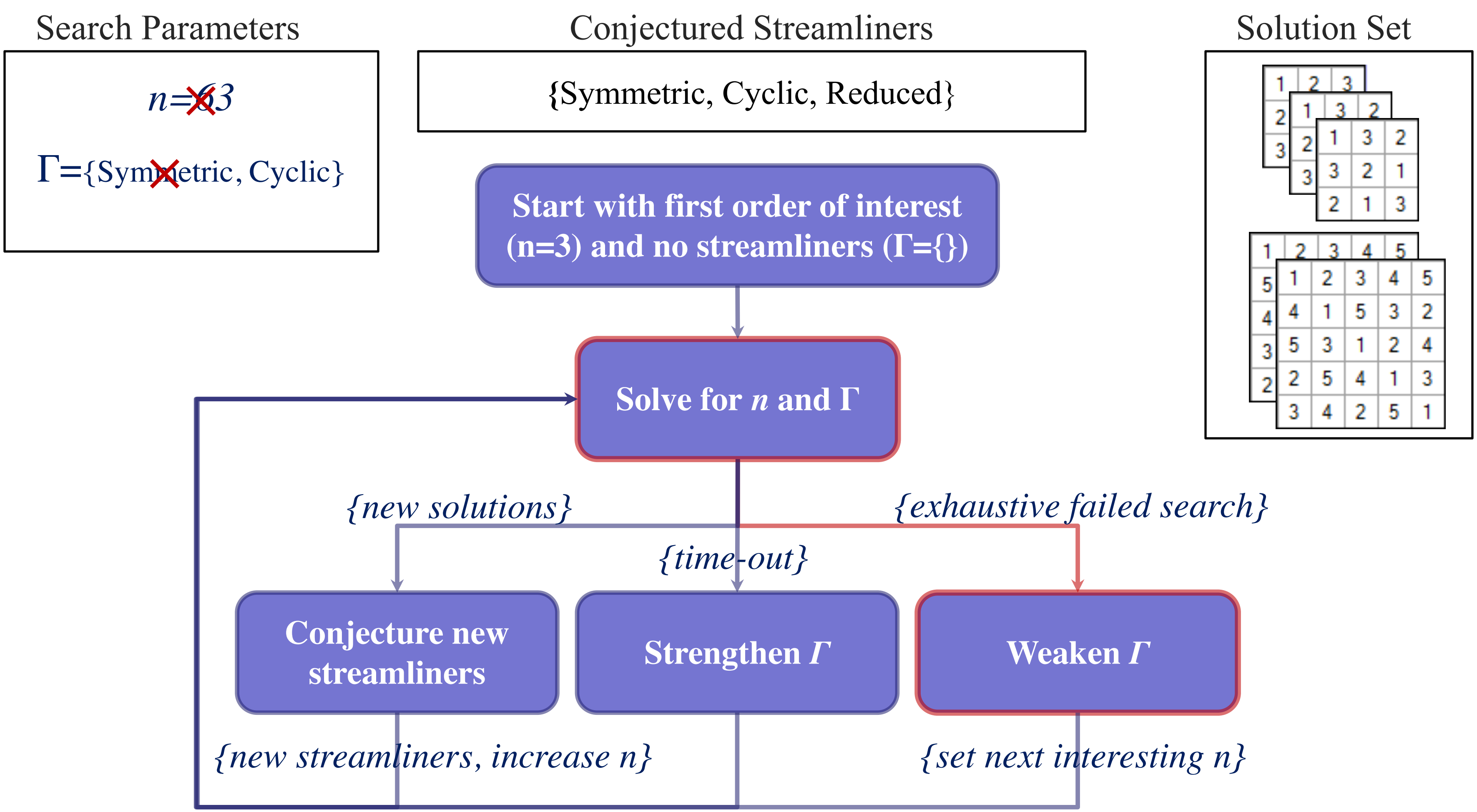
{Symmetric, Cyclic, Reduced}



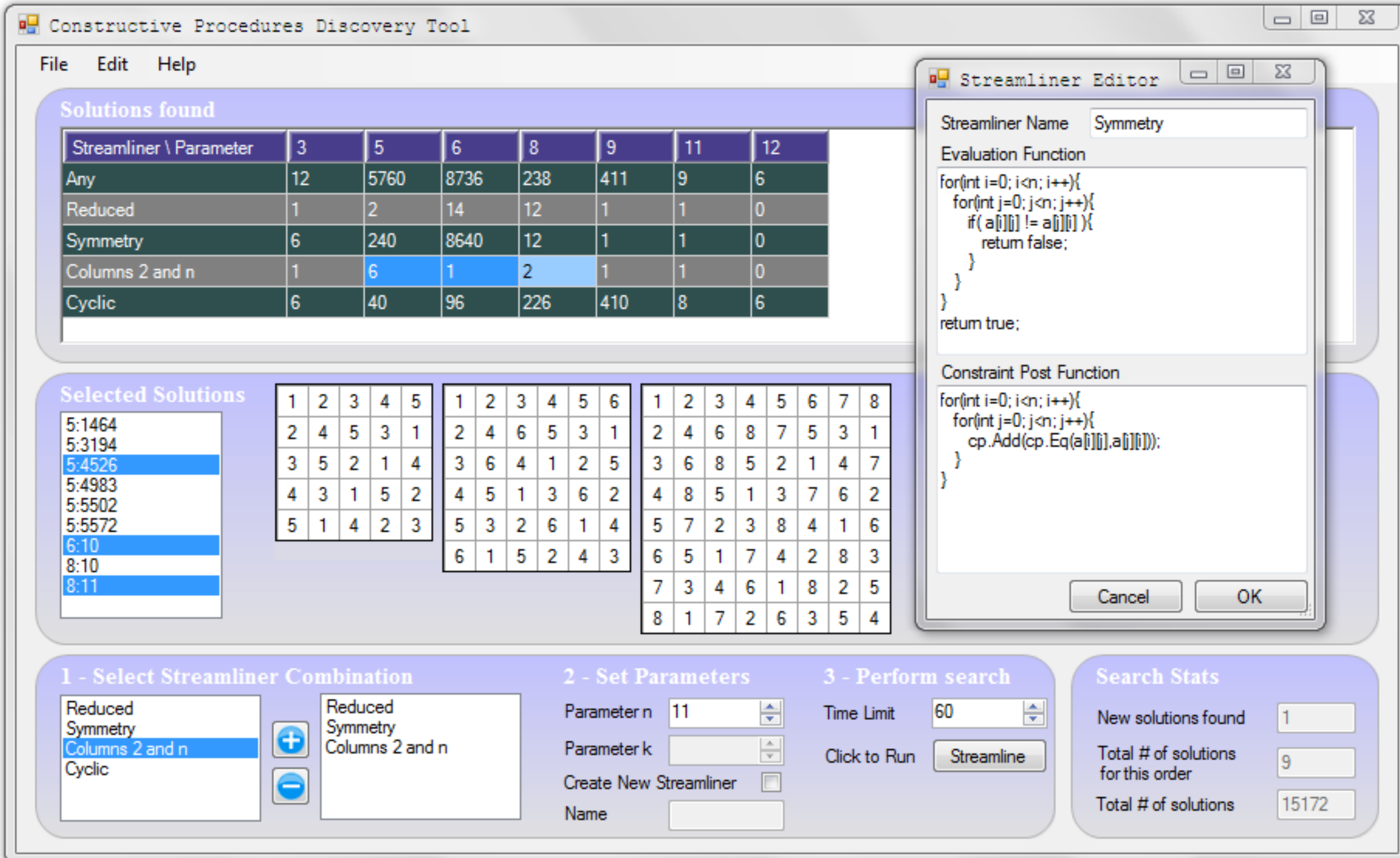
Solution Set



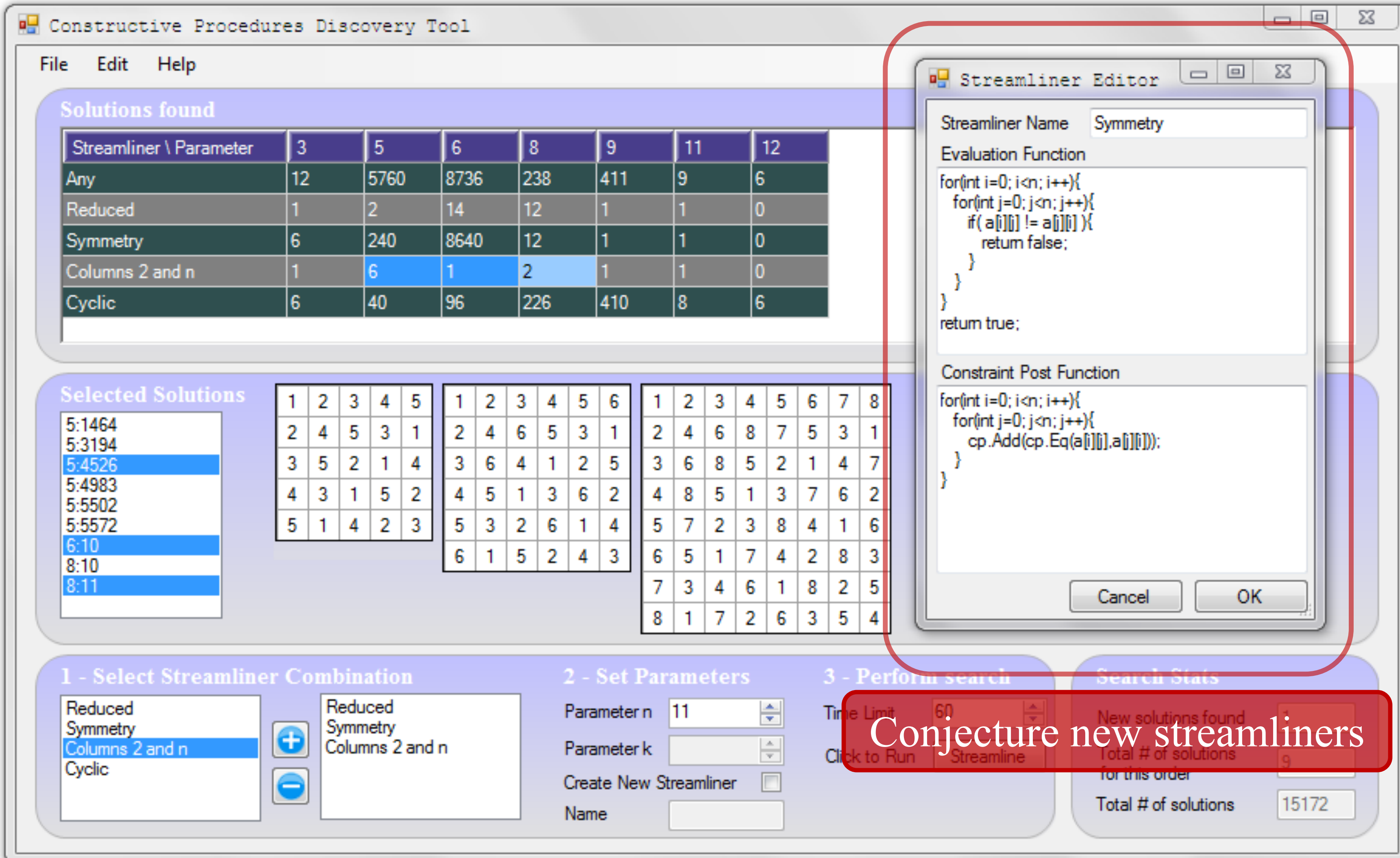
Overview on the *SBLS* problem



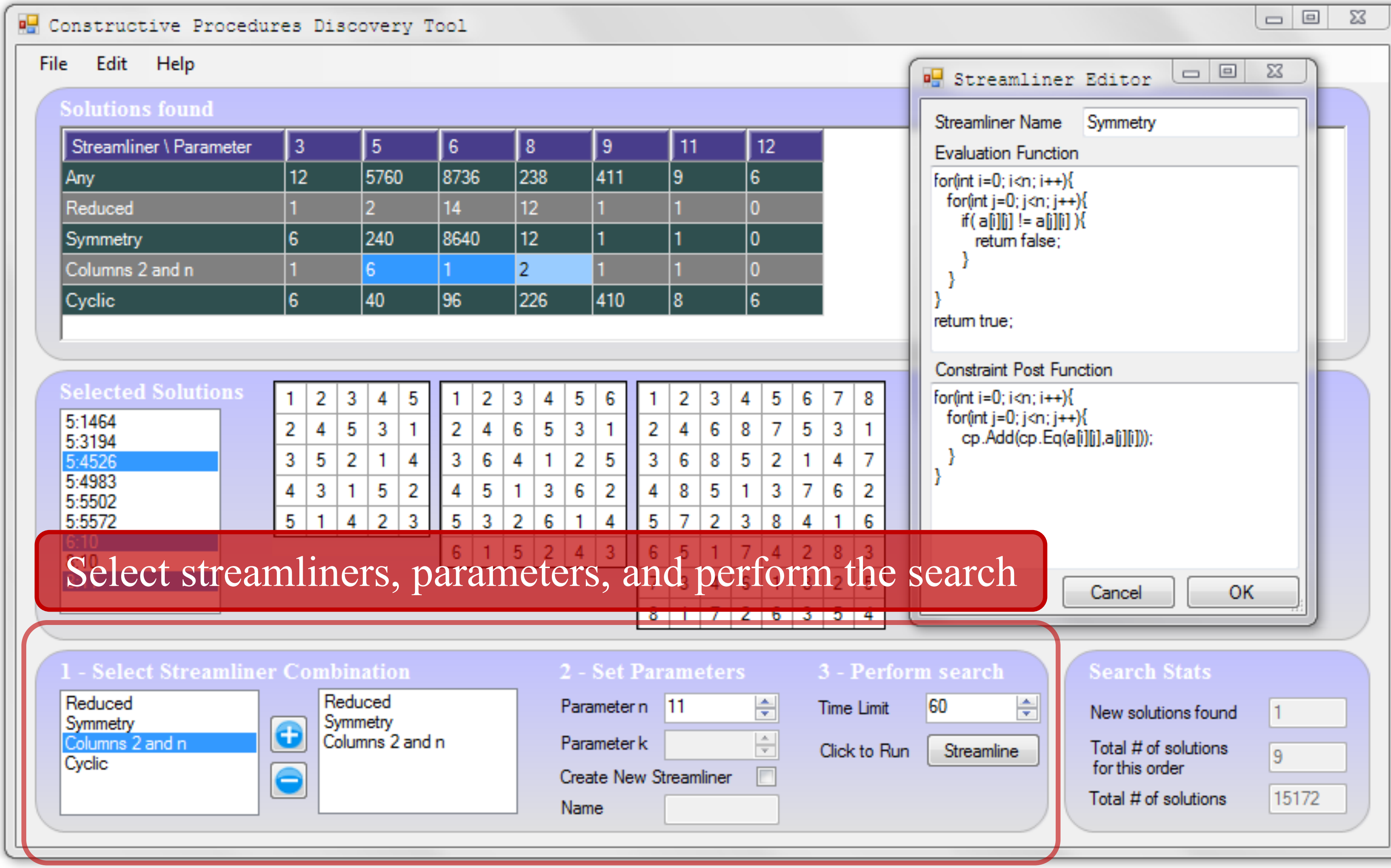
GUI for Human-guided Streamlined Search



GUI for Human-guided Streamlined Search

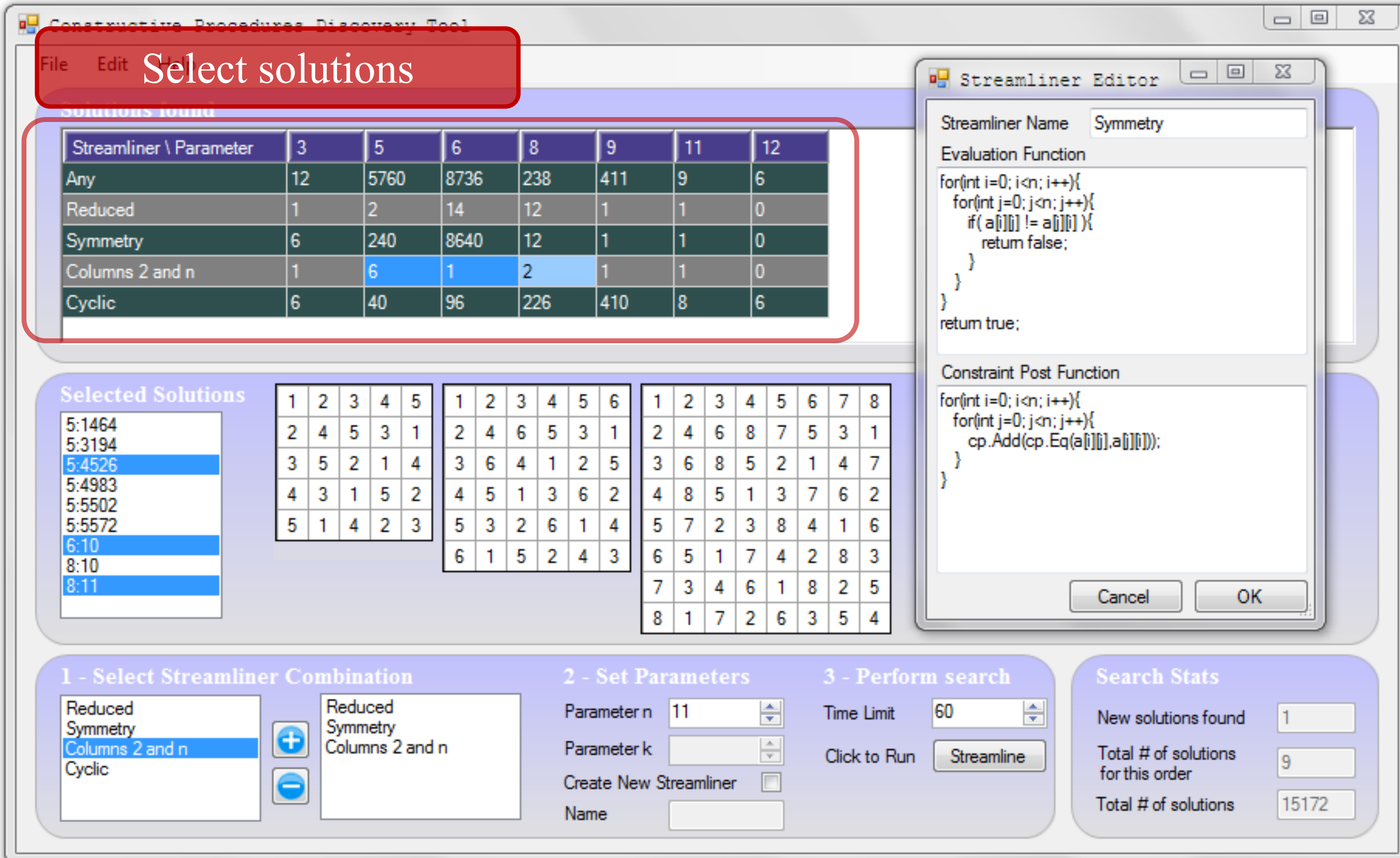


GUI for Human-guided Streamlined Search

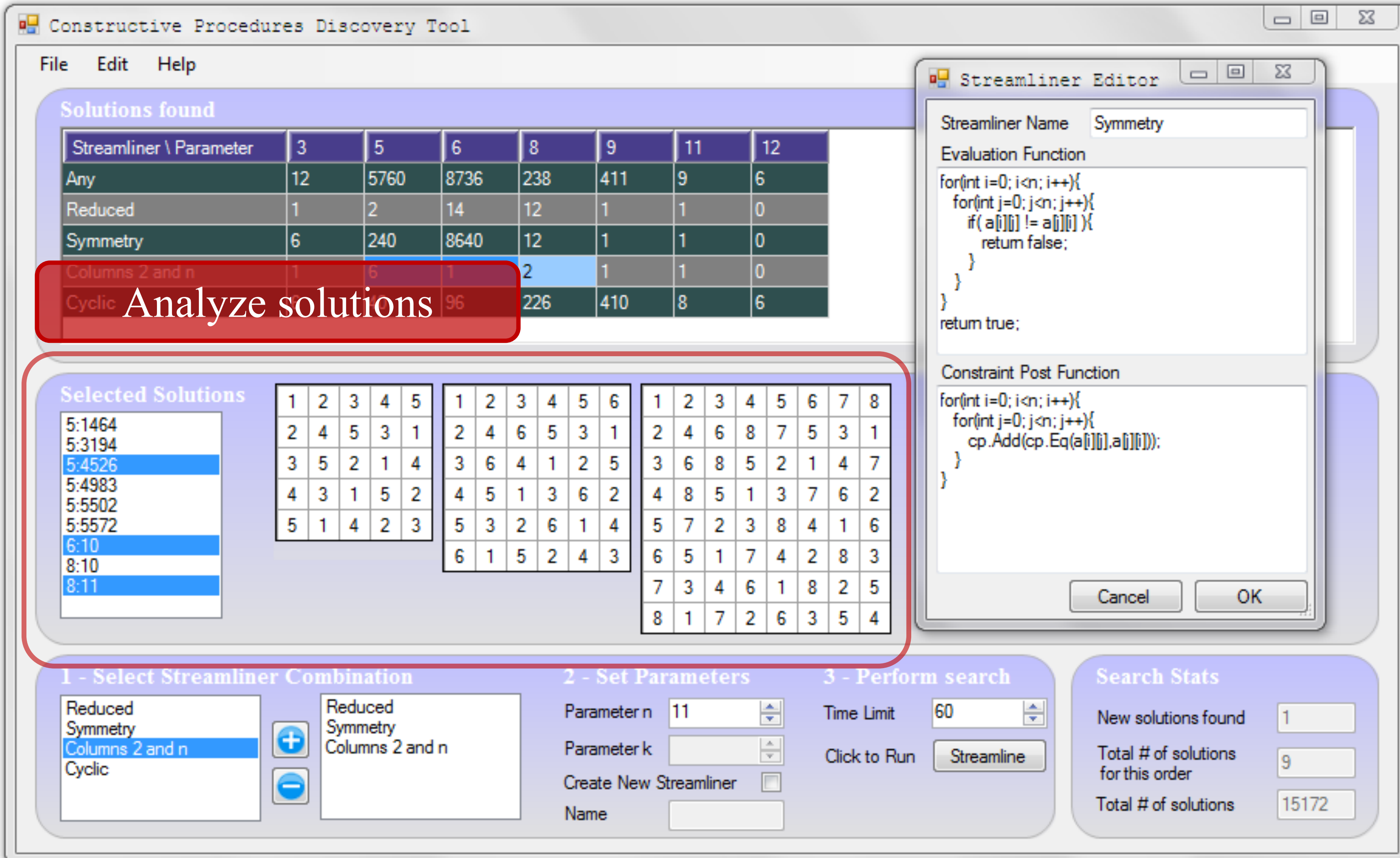


Select streamliners, parameters, and perform the search

GUI for Human-guided Streamlined Search



GUI for Human-guided Streamlined Search



Outline

- Motivation
- Example Domain
- Proposed Framework
- Application to the *Spatially-Balanced Latin square* problem
  - Successful Streamliners
  - Constructive Procedure 1
  - Constructive Procedure 2
- Application to the *Weak Schur Number* problem
- Conclusions and Future work

Application to the *SBLS* problem: Construction 1

Successful Key Streamliners:

{Diagonal symmetry, Reduced form, Assignments of columns 2 and n, Multiples of i in row i, Second sequence decreasing}

Streamliners	5	6	8	9	11	14
$\Gamma_1 = \emptyset$	**5760**	15878	-	-	-	-
$\Gamma_2 = \Gamma_1 \cup \{Symmetric\}$	**240**	8447	714	43	-	-
$\Gamma_3 = \Gamma_2 \cup \{Reduced\}$	**2**	**14**	**14**	51	-	-
$\Gamma_4 = \Gamma_3 \cup \{Columns\ 2\ \&\ \mathrm{n}\}$	**1**	**1**	**2**	**1**	**1**	-
$\Gamma_5 = \Gamma_4 \cup \{Multiples\ of\ i\}$	**1**	**1**	**2**	**1**	**1**	1

Fig: Number of SBLSs generated in 60 seconds, by order and streamliners (Bold indicates exhaustive search).

Application to the *SBLS* problem: Construction 1

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

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

1	2	3	4	5	6	7	8	9	10	11
2	4	6	8	10	11	9	7	5	3	1
3	6	9	11	8	5	2	1	4	7	10
4	8	11	7	3	1	5	9	10	6	2
5	10	8	3	2	7	11	6	1	4	9
6	11	5	1	7	10	4	2	8	9	3
7	9	2	5	11	4	3	10	6	1	8
8	7	1	9	6	2	10	5	3	11	4
9	5	4	10	1	8	6	3	11	2	7
10	3	7	6	4	9	1	11	2	8	5
11	1	10	2	9	3	8	4	7	5	6

1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	4	6	8	10	12	14	13	11	9	7	5	3	1
3	6	9	12	14	11	8	5	2	1	4	7	10	13
4	8	12	13	9	5	1	3	7	11	14	10	6	2
5	10	14	9	4	1	6	11	13	8	3	2	7	12
6	12	11	5	1	7	13	10	4	2	8	14	9	3
7	14	8	1	6	13	9	2	5	12	10	3	4	11
8	13	5	3	11	10	2	6	14	7	1	9	12	4
9	11	2	7	13	4	5	14	6	3	12	8	1	10
10	9	1	11	8	2	12	7	3	13	6	4	14	5
11	7	4	14	3	8	10	1	12	6	5	13	2	9
12	5	7	10	2	14	3	9	8	4	13	1	11	6
13	3	10	6	7	9	4	12	1	14	2	11	5	8
14	1	13	2	12	3	11	4	10	5	9	6	8	7

Application to the *SBLS* problem: Construction 1

1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	4	6	8	10	12	14	13	11	9	7	5	3	1
3	6	9	12	14	11	8	5	2	1	4	7	10	13
4	8	12	13	9	5	1	3	7	11	14	10	6	2
5	10	14	9	4	1	6	11	13	8	3	2	7	12
6	12	11	5	1	7	13	10	4	2	8	14	9	3
7	14	8	1	6	13	9	2	5	12	10	3	4	11
8	13	5	3	11	10	2	6	14	7	1	9	12	4
9	11	2	7	13	4	5	14	6	3	12	8	1	10
10	9	1	11	8	2	12	7	3	13	6	4	14	5
11	7	4	14	3	8	10	1	12	6	5	13	2	9
12	5	7	10	2	14	3	9	8	4	13	1	11	6
13	3	10	6	7	9	4	12	1	14	2	11	5	8
14	1	13	2	12	3	11	4	10	5	9	6	8	7

a	a+i

a	a-i

Approach reveals patterns. Final step is figuring out boundary conditions.

Application to the *SBLS* problem: Construction 1

1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	4	6	8	10	12	14	13	11	9	7	5	3	1
3	6	9	12	14	11	8	5	2	1	4	7	10	13
4	8	12	13	9	5	1	3	7	11	14	10	6	2
5	10	14	9	4	1	6	11	13	8	3	2	7	12
6	12	11	5	1	7	13	10	4	2	8	14	9	3
7	14	8	1	6	13	9	2	5	12	10	3	4	11
8	13	5	3	11	10	2	6	14	7	1	9	12	4
9	11	2	7	13	4	5	14	6	3	12	8	1	10
10	9	1	11	8	2	12	7	3	13	6	4	14	5
11	7	4	14	3	8	10	1	12	6	5	13	2	9
12	5	7	10	2	14	3	9	8	4	13	1	11	6
13	3	10	6	7	9	4	12	1	14	2	11	5	8
14	1	13	2	12	3	11	4	10	5	9	6	8	7

a	a+i

a	a-i

for *row* $i = 1, \dots, N$ **do**
  $k = 1;$ // Sequence number
  $j = 1;$ // Column index
  $a_{i,j} = i;$ // First symbol of row i
  while $j < N$ **do**
    if k *is odd* **then** // Odd sequence
      while $a_{i,j} + i \leq N$ *and* $j < N$ **do**
        $a_{i,j+1} = a_{i,j} + i;$
        $j = j + 1;$
    else // Even sequence
      while $a_{i,j} - i \geq 1$ *and* $j < N$ **do**
        $a_{i,j+1} = a_{i,j} - i;$
        $j = j + 1;$
    if $j < N$ **then** // Switch sequence
      if k *is odd* **then**
        $a_{i,j+1} = 2N + 1 - i - a_{i,j};$
      else
        $a_{i,j+1} = i - a_{i,j};$
      $k = k + 1;$
      $j = j + 1;$

Algorithm : `SBLS-sequence` procedure for SBLS of order N, when $2N + 1$ is prime.

Application to the *SBLS* problem: Construction 1

1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	4	6	8	10	12	14	13	11	9	7	5	3	1
3	6	9	12	14	11	8	5	2	1	4	7	10	13
4	8	12	13	9	5	1	3	7	11	14	10	6	2
5	10	14	9	4	1	6	11	13	8	3	2	7	12
6	12	11	5	1	7	13	10	4	2	8	14	9	3
7	14	8	1	6	13	9	2	5	12	10	3	4	11
8	13	5	3	11	10	2	6	14	7	1	9	12	4
9	11	2	7	13	4	5	14	6	3	12	8	1	10
10	9	1	11	8	2	12	7	3	13	6	4	14	5
11	7	4	14	3	8	10	1	12	6	5	13	2	9
12	5	7	10	2	14	3	9	8	4	13	1	11	6
13	3	10	6	7	9	4	12	1	14	2	11	5	8
14	1	13	2	12	3	11	4	10	5	9	6	8	7

a	a+i

a	a-i

for *row* $i = 1, \ldots, N$ **do**
  $k = 1;$ // Sequence number
  $j = 1;$ // Column index
  $a_{i,j} = i;$ // First symbol of row i
  while $j < N$ **do**
    if k *is odd* **then** // Odd sequence
      while $a_{i,j} + i \leq N$ *and* $j < N$ **do**
        $a_{i,j+1} = a_{i,j} + i;$
        $j = j + 1;$
    else // Even sequence
      while $a_{i,j} - i \geq 1$ *and* $j < N$ **do**
        $a_{i,j+1} = a_{i,j} - i;$
        $j = j + 1;$
    if $j < N$ **then** // Switch sequence
      if k *is odd* **then**
        $a_{i,j+1} = 2N + 1 - i - a_{i,j};$
      else
        $a_{i,j+1} = i - a_{i,j};$
      $k = k + 1;$
      $j = j + 1;$

Algorithm : `SBLS-sequence` procedure for SBLS of order N, when $2N + 1$ is prime.

Proof of Correctness in [R. Le Bras, et al., *Polynomial Time Construction for Spatially Balanced Latin Squares*, 2012]

Application to the *SBLS* problem: Construction 2

1	2	6	3	9	7	13	4	11	10	12	8	5	14
2	3	7	4	10	8	14	5	12	11	13	9	6	1
3	4	8	5	11	9	1	6	13	12	14	10	7	2
4	5	9	6	12	10	2	7	14	13	1	11	8	3
5	6	10	7	13	11	3	8	1	14	2	12	9	4
6	7	11	8	14	12	4	9	2	1	3	13	10	5
7	8	12	9	1	13	5	10	3	2	4	14	11	6
8	9	13	10	2	14	6	11	4	3	5	1	12	7
9	10	14	11	3	1	7	12	5	4	6	2	13	8
10	11	1	12	4	2	8	13	6	5	7	3	14	9
11	12	2	13	5	3	9	14	7	6	8	4	1	10
12	13	3	14	6	4	10	1	8	7	9	5	2	11
13	14	4	1	7	5	11	2	9	8	10	6	3	12
14	1	5	2	8	6	12	3	10	9	11	7	4	13

$c_{1,1} = 1;$ // Generate 1st row of the conjugate
for *column* $j = 2, \ldots, N$ **do** // Observed pattern $1, 2, 4, \ldots$
  if $2c_{1,j-1} \leq N$ **then**
    $c_{1,j} = 2c_{1,j-1};$
  else
    $c_{1,j} = 2N + 1 - 2c_{1,j-1};$
for *row* $i = 2, \ldots, N$ **do** // Subsequent rows
  $c_{i,1} = c_{i-1,N};$ // Shifted version of previous
  for *column* $j = 2, \ldots, N$ **do**
    $c_{i,j} = c_{i-1,j-1};$
for *row* $i = 1, \ldots, N$ **do** // Generate SBLS from conjugate
  for *column* $j = 1, \ldots, N$ **do**
    $a_{i,c_{i,j}} = j;$

Algorithm: `SBLS-Cyclic` procedure.

Observation I

Incremental approach in terms of size is necessary:

Need to reveal structure by generating increasingly larger solutions using increased streamlining (and thus, structure).

True solution structure only becomes visible around order 14.

At that size

(1) Can’t generate any solution of that size without strong streamlining,

(2) But, even if we could, arbitrary solutions will not show structure.

Observation II

We conjecture that many hard combinatorial design problems have effective constructive procedures. (E.g., quasigroup existence problems, code design, Ramsey graphs (numbers), van der Waerden, Schur numbers etc.).

So far, such problems have been tackled with combinatorial search (SAT solvers, equational thm. provers) on a per instance basis. Does resolve open questions in combinatorial design (e.g. new lower-bounds) but does not reveal the general solution structure.

E.g. Slaney et al. 1993, Herwig et al. 2007, Heule and Walsh 2010 Eliahou et al. 2012, Fujita et al. 2013.

Example: the *Weak Schur* problem

Problem Definition:

- A set is ***(weakly)* sum free** if for any two *(distinct)* elements of this set, their sum does not belong to the set.
- The ***Weak Schür Number*** of order k, $WS(k)$, is the largest integer n for which there exists a partition of $[1,n]$ into k weakly sum-free sets.

k=3 sets

1 2 4 8 11 22
3 5-7 19 21 23
9 10 12-18 20

Each of the 3 sets is such that, for any 2 elements of the set, their sum does not belong to the same set.

Fig: **Partition of [1,23] into 3 weakly sum-free sets, proving $WS(3) \geq 23$**

Application to the *Weak Schur* problem

Best known lower bounds:

Approach	***WS*(5)**	***WS*(6)**	**Reference**
(not disclosed)	196	-	[G.W. Walker, AMM’50]
Theoretical bound *(not proved)*	188	554	[J.H. Braun, AMM’50]
SAT	196	572	[Eliahou et al., Computers & Math Applications’12]
Multi-level Tabu-Search	196	574	[Fonlupt et al., EA’11]
SAT *(no certificate)*	196	575	[Eliahou et al., Computers & Math Applications’12] *(revised)*

New: $WS(6) \geq 581$

CORNELL UNIVERSITY

Successful Key Streamliners:

{Ordered sets, constrained minimum of each set, partial assignments, sequences of consecutive integers, sequence interleaving}

Application to the *Weak Schur* problem

Successful Key Streamliners:

{Ordered sets, constrained minimum of each set, partial assignments, sequences of consecutive integers, sequence interleaving}

1 2 4 8 11 22 25 50 63 68 139 149 154 177 182 192 198 393 398 408 413 436 450 455 521 526 540 563 568 578
3 5-7 19 21 23 51-53 64-66 136-138 150-152 179-181 193-195 395-397 409-411 438-440 451-453 523-525 536-538 565-567 579-581
9 10 12-18 20 54-62 140-148 183-191 399-407 441-449 527-535 569-577
24 26-49 153 155-176 178 412 414-435 437 539 541-562 564
67 69-135 454 456-520 522
196 197 199-392 394

Partition of [1,581] into 6 weakly sum-free sets, proving *WS*(6) ≥ 581.

(le Bras, Gomes, Selman AAAI 2012)

- Incrementally obtained, although **not** an example of a **fully constructive procedure** yet, any progress on Schur numbers is quite **significant** given their long history.

Summary

- **Structure discovery** through a general framework that integrates **streamlined** combinatorial search with **human insight and intuition** in an **iterative** approach to discover **efficient constructive procedures**.

- Provides the **first constructive procedures** for the ***Spatially-Balanced Latin Square*** problem.

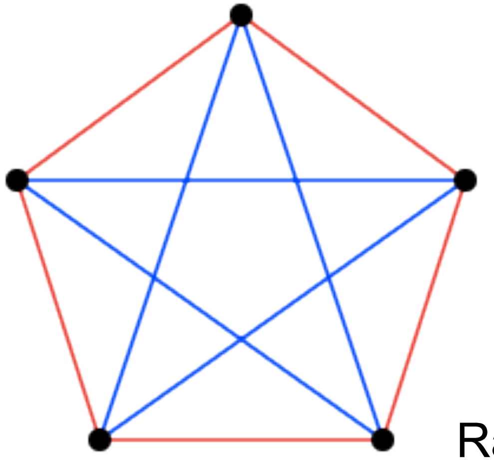
- Also, improves the best known **lower bound** for the ***Weak Schur Number*** problem, and, most recently, for "Graceful graphs."

- Extensions **crowd-source** the search for regularities in the solution sets (on Mechanical Turk).

And much, much more ambitiously…

--- Ramsey numbers (Erdos), and

--- possibly, new types of algorithms.



here

5 nodes

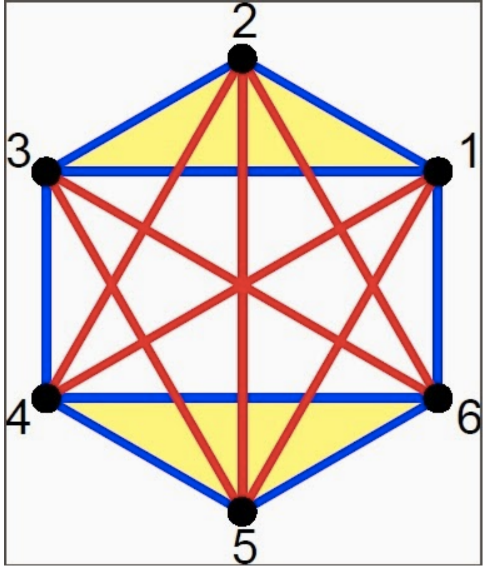
red/blue edges

no blue or red triangle

6+ nodes

always a blue or red triangle

R(3,3) = 6



Ramsey theory about "forced" structure

"unavoidable structure in our universe"

R(3,3) = 6

R(4,4) = 18

R(5,5) = ?

Erdos: ***"Imagine an alien force, vastly more powerful than us landing on Earth and demanding the value of R(5, 5) or they will destroy our planet." [hmm?]***

In that case, we should marshal all our computers and all our mathematicians and attempt to find the value. …

But suppose, instead, that they asked for* R*(6, 6), then we should attempt to destroy the aliens".

Situation is probably not quite as dire. ☺

The key will be to find the hidden structure in the solution space.

Part II:
Crowdsourcing for Backdoor Variables:
Materials Discovery Domain

Briefly!

Intelligent Information
Systems Institute

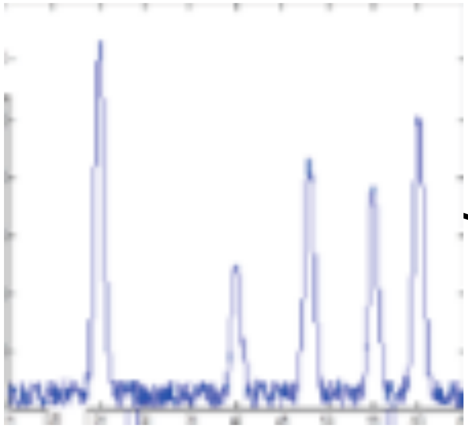
Institute for Computational Sustainability

ICS

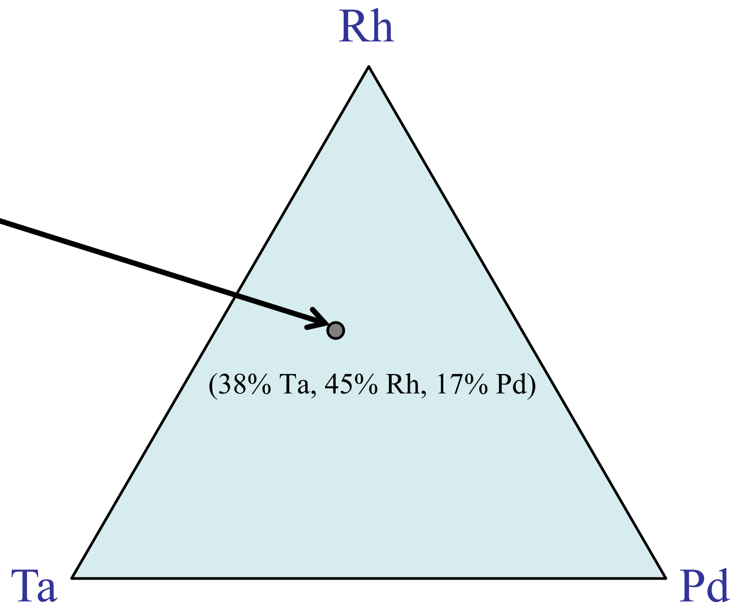
Warning: Real-world, noisy data ahead

Motivation

Study of new materails:
Identifying crystalline structure using **X-Ray Diffraction patterns**



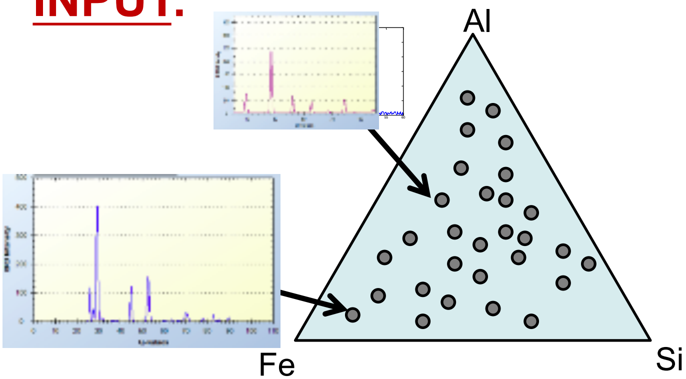
[Source: *Pyrotope*, Sebastien Merkel]



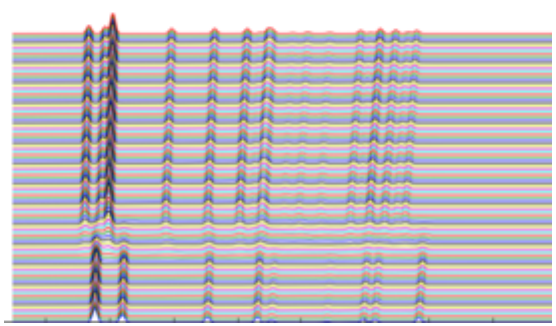
With Caltech, working on screening 1 million samples per day.

Phase Map Identification Problem

INPUT:



Collection of X-Ray diffraction Patterns. **Noisy and hidden structure.**



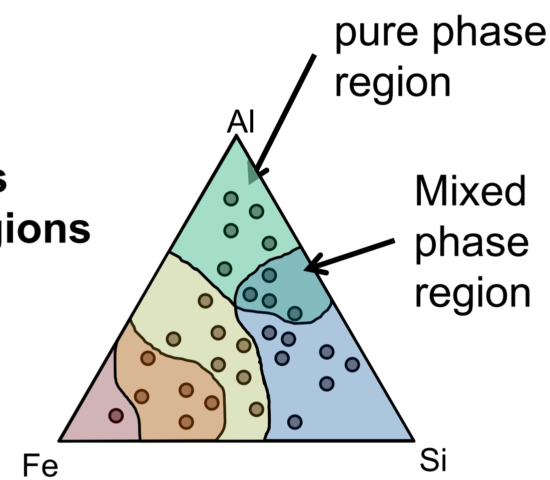
Additional Physical Requirements:

- Phase Connectivity
- Mixtures of ≤ 3 pure phases
- Peaks shift by ≤ 15% within a region
  - Continuous and Monotonic
- Small peaks might be discriminative
- Peak locations matter,
  more than peak intensities

OUTPUT:

m **phase regions**

- ***k*** **pure regions**
- ***m-k*** **mixed regions**



Complex Full Constraint Optimization Model

Variables	Description
p_{ki}	Normalizing element for phase k in pattern P_i
a_{ki}	Whether phase k is present in pattern P_i
q_k	Set of normalized peak locations of phase B_k

$$(a_{ki} = 0) \iff (p_{ki} = 0) \qquad \forall\, 1 \le k \le K, 1 \le i \le n \tag{1}$$

$$1 \le \sum^{K} a_{si} \le M \qquad \forall\, 1 \le i \le n \tag{2}$$

$$(p_{ki} = j) \Rightarrow (q_k \subseteq r_{ij}) \qquad \forall\, 1 \le k \le K, 1 \le i \le n, 1 \le j \le |P_i| \tag{3}$$

$$(p_{ki} = j \wedge \sum_{s=1}^{K} a_{si} = 1) \Rightarrow (r_{ij} \subseteq q_k)$$
$$\forall\, 1 \le k \le K, 1 \le i \le n, 1 \le j \le |P_i| \tag{4}$$

$$(p_{ki} = j \wedge p_{k'i} = j' \wedge \sum_{s=1}^{K} a_{si} = 2)$$
$$\Rightarrow (member(r_{ij}[j''], q_k) \vee member(r_{ij'}[j''], q_{k'}))$$
$$\forall\, 1 \le k, k' \le K, 1 \le i \le n, 1 \le j, j', j'' \le |P_i| \tag{5}$$

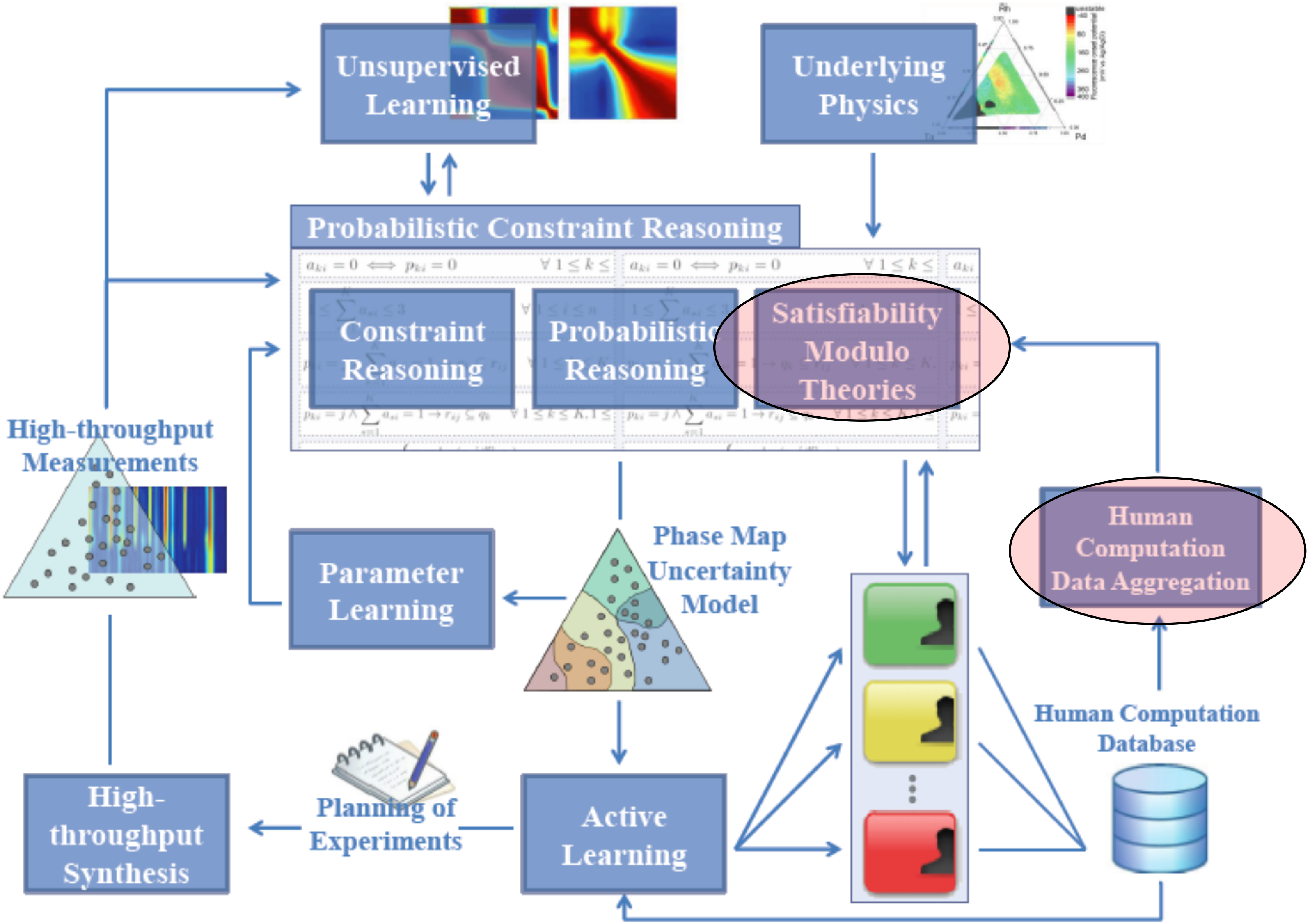
$$(p_{ki} = j) \Rightarrow (p_{ki'} \ne j') \qquad \forall\, 1 \le k \le K, (i, j, i', j') \in \Phi \tag{6}$$

$$\Phi = \{(i, j, i', j') \mid \frac{P_i[j]}{P'_i[j']} < 1/\delta \vee \frac{P_i[j]}{P'_i[j']} > \delta, i < i'\}$$

$$basisPatternConnectivity(\{a_{ki} | 1 \le i \le n\}) \qquad \forall\, 1 \le k \le K \tag{7}$$

Drawback:

Does not scale to full-scale realistic instances; poor propagation of experimental noise.


Unsupervised Learning
Underlying Physics
Probabilistic Constraint Reasoning
Constraint Reasoning
Probabilistic Reasoning
Satisfiability Modulo Theories
High-throughput Measurements
Parameter Learning
Phase Map Uncertainty Model
Human Computation Data Aggregation
Human Computation Database
High-throughput Synthesis
Planning of Experiments
Active Learning

SAT Modulo Theory (SMT) Approach: Experimental Validation

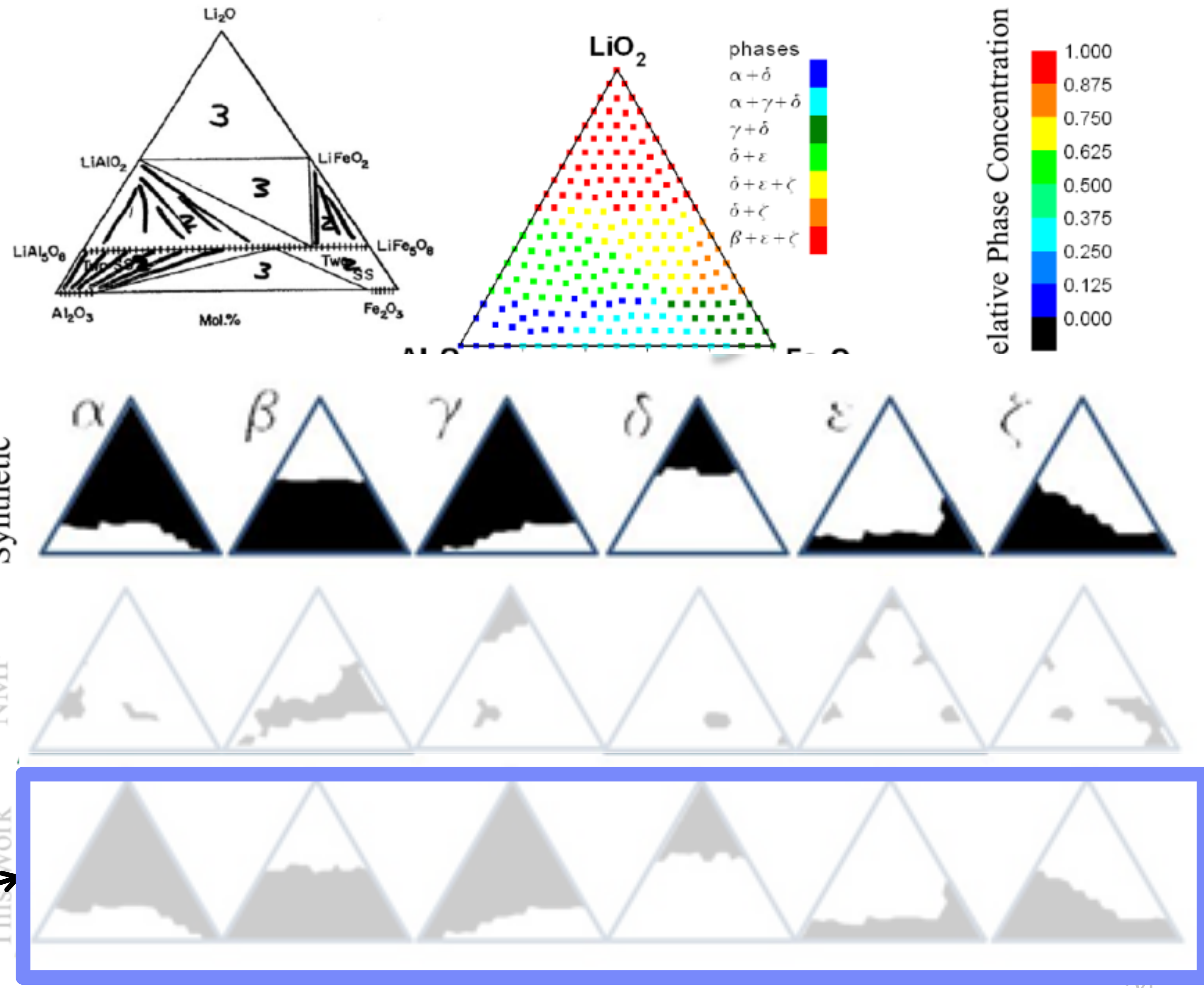
Al-Li-Fe instance with 6 phases:

Ground truth (known)

Previous work

(ML Based approaches) violates many physical requirements

Our SMT approach is much more robust

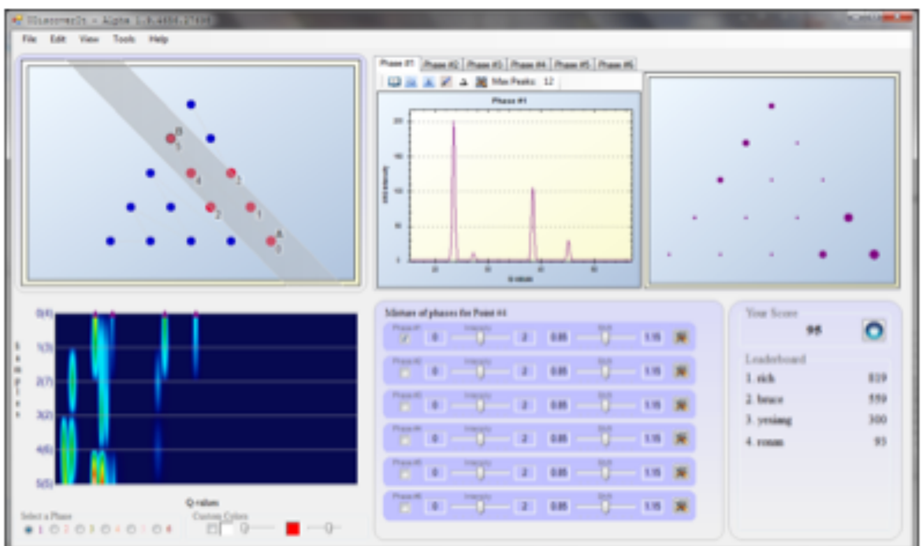


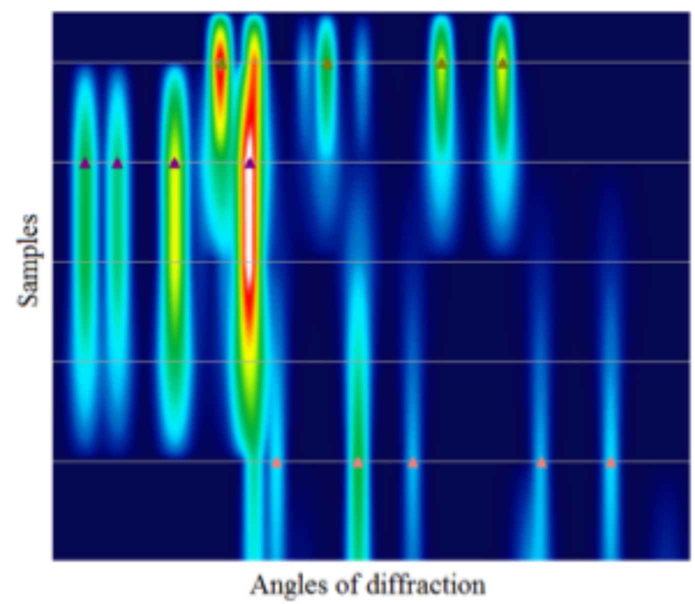
Institute for Computational Sustainability ICS

UDiscoverIt:

- Can human input boost the performance of combinatorial reasoning and optimization methods (SMT)?
- Human input about potential grouping of diffraction peaks collected on Amazon’s Mechanical Turk. (About 500 patterns)

Provides info on potential so-called backdoor variables.





System	Dataset P	L^*	K	#var	#cst	Time w/o user input (s)	Time w/ user input (s)	# assigned var. by user
A/B/C	36	8	4	408	2095	3502	**150**	19 (4.6%)
A/B/C	60	8	4	624	3369	17345	**261**	18 (2.9%)
Al/Li/Fe	15	6	6	267	1009	79	**27**	6 (2.2%)
Al/Li/Fe	28	6	6	436	1864	346	**83**	12 (2.7%)
Al/Li/Fe	28	8	6	490	2131	10076	**435**	26 (5.3%)
Al/Li/Fe	28	10	6	526	2309	28170	**188**	23 (4.3%)
Al/Li/Fe	45	7	6	693	3281	18882	**105**	28 (4.0%)
Al/Li/Fe	45	8	6	711	3410	46816	**74**	30 (4.2%)

Human input can dramatically speed up the performance of combinatorial optimization methods (identification of backdoor variables).

Leverages the complementary strength of human input, providing global insights into problem structure, and the power of combinatorial solvers to exploit complex constraints.

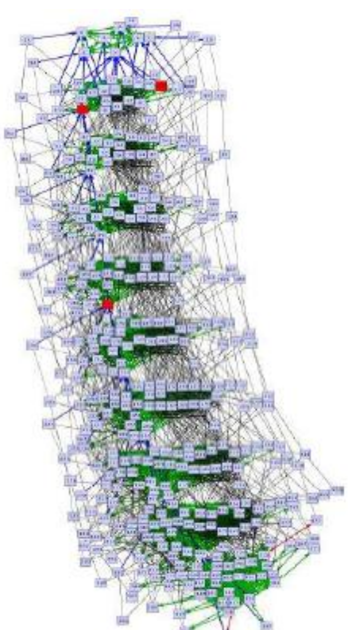
Note: Human input identifies subsets of peaks that (quite likely) go together.

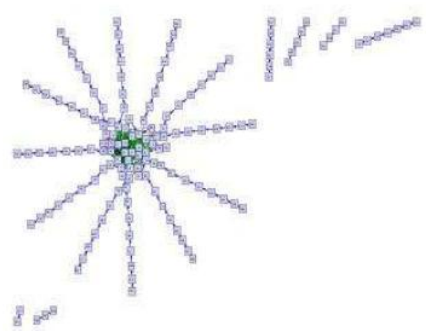
Observation

Visual representation allows human input to be effective in discovering structure and setting backdoor variables in this domain.

Contrast with earlier work:

SAT encodings of planning problems. It was very difficult to give any semantic interpretation to the backdoor variables of the problem instances. Synthetic planning formulas with log(n) size backdoors (provably).





Setting 2 out of 392 variables.

(Hoffmann, Gomes, Selman AIPS-06)

Conclusions

We have discussed how human insights can be combined with reasoning and optimization methods for scientific discovery.

Finite Mathematics --- Obtained first procedure for *Spatially Balanced Latin Square* construction.

Materials Discovery --- Boosted interpretation of X-ray diffraction patterns with crowdsourced input.