From Streamlined Combinatorial Search to Efficient Constructive Procedures

Ronan Le Bras, Carla P. Gomes, Bart Selman

Motivation

- Significant progress in the area of **search, constraint satisfaction and automated reasoning** driven by challenge problems in combinatorics.

- One shortcoming is that these methods do not provide **further mathematical insights**, as they are, in essence, a form of proof by demonstration.

- We propose a framework that combines specialized combinatorial search (***streamlining***) with **human insights**, in a complementary, iterative approach.

- Ultimately, the goal is to discover **efficient constructive procedures** (polynomial-time algorithms that take as input a size parameter, N, and generate a certain combinatorial object of size N).

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

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

Example Domains

Spatially-Balanced Latin Square (SBLS) Problem:

						(1,2)	(1,3)	(2,4)	(2,6)	(5,6)
1	2	3	4	5	6	1	2	2	4	1
2	4	6	5	3	1	5	1	1	1	1
3	6	4	1	2	5	1	3	2	3	4
4	5	1	3	6	2	3	1	5	1	3
5	3	2	6	1	4	2	3	3	1	3
6	1	5	2	4	3	2	4	1	3	2
Total Row Distance (pair)						14	14	14	14	14
Average Row Distance (pair)						2.33	2.33	2.33	2.33	2.33

(Pair Symbol Distance (per row))

Figure: Spatially balanced Latin square of order 6

- Each symbol from 1 to n appears exactly once in each row and column *(Latin Square structure)*.

- The average distance (column-wise) of a pair of symbols is the same for any pair *(Balanced structure)*.

- Computationally challenging combinatorial design problem with applications in the area of **crop rotation** and **drug design**.

Weak Schur Number (WS) Problem:

Table: Partition of $[1, 23]$ into 3 weakly sum-free sets, proving $WS(3) \geq 23$ ('5-7' means '5 6 7' are in the set). Each of the 3 sets is such that the sum of any two of its members is not in the set.

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

- Schur Numbers are closely related to **Ramsey theory**, and are a **notoriously hard** area of combinatorics.

- A set is **weakly sum free** if for any two elements of this set, their sum does not belong to the set.

- The **Weak Schur 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.

Discover-Construction Procedure

Overview of the proposed strategy:

1) **Analyze** smaller size solutions, and **conjecture potential regularities** in the solutions.

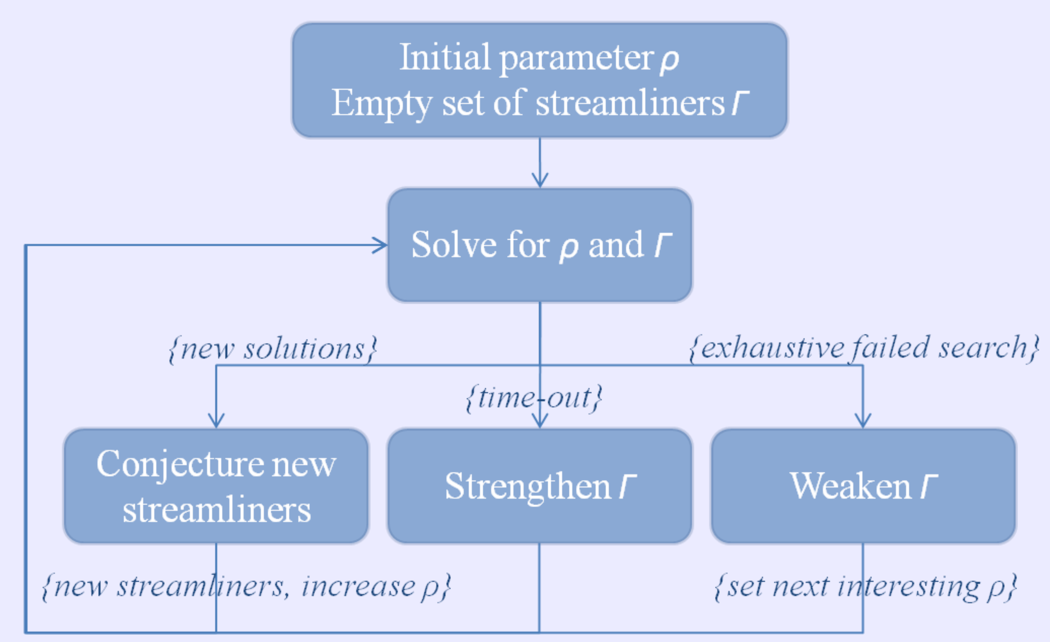
2) Validate through **streamlining** the observed regularities.

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**.

```
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.



GUI for Human-guided Streamlined Search

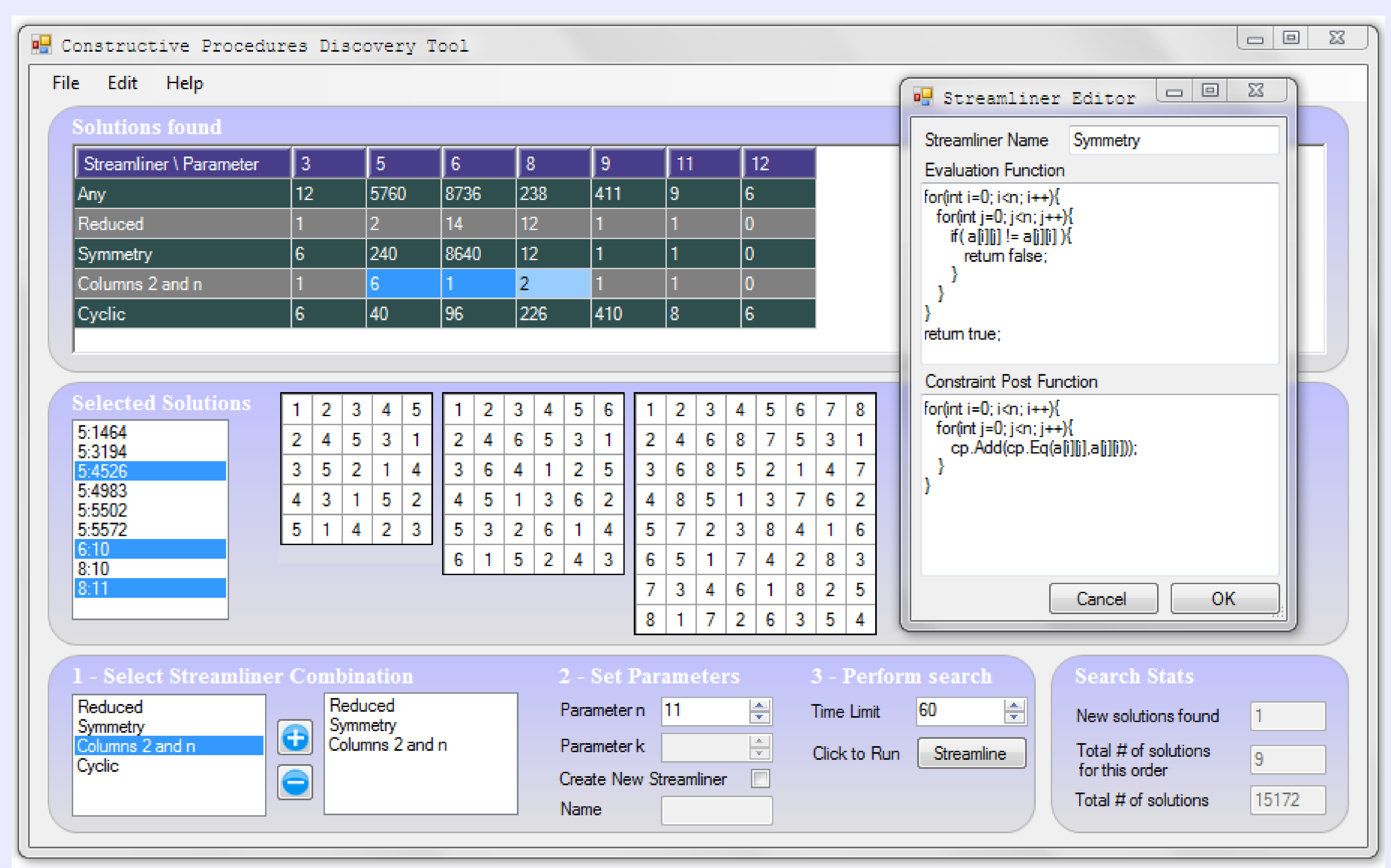


Figure: User interface for human-guided streamlined search to discover constructive procedures

Results on the *SBLS* Problem

Successful Key Streamliners:

{Diagonal symmetry, reduced form, assignments of columns 2 and *n*, multiples of *i* in row *i*, second sequence decreasing}

Table: Number of SBLSs generated by size and streamliner. A 60-second time-out was used. Bold indicates exhaustive search.

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	-	-
$\Gamma_4 = \Gamma_3 \cup \{Columns\ 2\ \&\ n\}$	**1**	**1**	**2**	**1**	**1**	-
$\Gamma_5 = \Gamma_4 \cup \{Multiples\ of\ i\}$	**1**	**1**	**2**	**1**	**1**	**1**

```
for row i = 1, ..., 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 ≤ N and j < N do
                a_{i,j+1} = a_{i,j} + i;
                j = j + 1;
        else                                // Even sequence
            while a_{i,j} − i ≥ 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.

- We introduce the **first constructive procedure** for ***Spatially Balanced Latin Squares.***

- The **largest SBLS known** to exist was of **order 35** and took about **2 weeks** of computation. Our algorithm generates a SBLS of **order 999** in **0.01 second**.

- Our constructive procedure confirms a **2004 conjecture** on the **existence** of arbitrary large SBLSs and of an **effective** way of constructing them.

Results on the *WS* Problem

Successful Key Streamliners:

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

Table: Partition of $[1, 581]$ into 6 weakly sum-free sets, proving $WS(6) \geq 581$. ('5-7' means '5 6 7' are in the set). Each of the six sets is such that the sum of any two of its members is not in the set.

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

- We provide a **new lower-bound** for the ***Weak Schur Numbers***, proving **WS(6) ≥ 581**.

- The **best known lower-bound** was **WS(6) ≥ 575**, found by (Eliahou et al., 2012) and improving on the '**WS(6) ≥ 574**' result of (Fonlupt et al., 2011)

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

Acknowledgments

The authors gratefully acknowledge the support of the **National Science Foundation**, award number 0832782.