

From Streamlined Combinatorial Search to Efficient Constructive Procedures



Ronan Le Bras Carla P. Gomes Bart Selman

Computer Science Computer Science Computer Science

July 26, 2012

AAAI'12

Motivation



Background:

Significant progress in the area of **search**, **constraint satisfaction**, and **automated reasoning**.

These approaches have been evaluated on problems such as:







N-Queens

Round-Robin Tournament

Orthogonal Latin squares



Motivation



Background:

Significant progress in the area of **search**, **constraint satisfaction**, and **automated reasoning**.

These approaches have been evaluated on problems such as:







N-Queens

Round-Robin Tournament

Orthogonal Latin squares

Yet, 1) constructions have been found by hand for these problems , and

2) these techniques do not provide real **mathematical insights** on the structure of the problem and how to devise general construction rules.

Goal: Design a framework to discover **efficient constructive procedures.**





In the context of **sustainability**, assume we have **4 fertilizers**, and we want to minimize their **impact on the 'dead-zones'** in the Gulf of Mexico.



Nitrogen Based Fertilizers

Dead zones





In the context of **sustainability**, assume we have **4 fertilizers**, and we want to minimize their **impact on the 'dead-zones'** in the Gulf of Mexico.



Nitrogen Based Fertilizers

Dead zones

In a more **general** context, assume we have **4 agronomic treatments** for growing beans and we want to **assess their effectiveness.**





In the context of **sustainability**, assume we have **4 fertilizers**, and we want to minimize their **impact on the 'dead-zones'** in the Gulf of Mexico.



Nitrogen Based Fertilizers

Dead zones

In a more **general** context, assume we have **4 agronomic treatments** for growing beans and we want to **assess their effectiveness.**

1) We need to **distribute** the treatments **evenly** over the test plots.







In the context of **sustainability**, assume we have **4 fertilizers**, and we want to minimize their **impact on the 'dead-zones'** in the Gulf of Mexico.



Nitrogen Based Fertilizers

Dead zones

In a more **general** context, assume we have **4 agronomic treatments** for growing beans and we want to **assess their effectiveness.**

1) We need to **distribute** the treatments **evenly** over the test plots.



2) We need to eliminate the correlation bias as much as possible, andNELL factor in the geometry of the fields

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).

SBLS of order 6





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).

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



Row Distance for pairs of colors





A computationally challenging combinatorial design problem:

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]





The largest SBLS ever found (35x35)



A computationally challenging combinatorial design problem:

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]



Conjecture [Gomes, Sellmann et al., CPAIOR'04]

There exist **arbitrary large SBLSs**, and an **effective** way of **constructing** them.



The largest SBLS ever found (35x35)



A computationally challenging combinatorial design problem:

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]



Conjecture [Gomes, Sellmann et al., CPAIOR'04]

There exist **arbitrary large SBLSs**, and an **effective** way of **constructing** them.

Goal: Discover an efficient construction.



The largest SBLS ever found (35x35)

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





- 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





Goal:

Exploit the **structure of some solutions** to dramatically **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 **regular ones**.



Proposed Framework: Overview of Streamlined Search





Strong **branching mechanisms** (by adding constraints based on **structure properties**) at **high levels** of the search tree. [Gomes and Sellmann, CP'04]







Recognizing Patterns and Regularities:



[Source: Marijn J.H. Heule, 2009]

Correcting Irregularities:



Generalizing / Formalizing Regularities:

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

Cyclic Latin square of order 3

Cyclic Latin square of order 4



Proposed Framework: Formal Description and Overview



 $\mathcal{O} \leftarrow \emptyset$: // Conjectured streamliners $\Gamma \leftarrow \emptyset$: // Search streamliners // Search parameter $\rho \leftarrow \rho_0;$ $\mathcal{S} \leftarrow \emptyset$: // Solutions found $\tau \leftarrow false;$ // Timeout flag repeat $Solve(P_{\rho}, \Gamma, t) \rightarrow (S', \tau);$ // Search for new solutions if $\mathcal{S}' \cap \mathcal{S} \neq \emptyset$ then $S \leftarrow S \cup S'$: // Case 1: successful search Analyze(S) $\rightarrow O'$; // 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 $\Gamma'' \subset \Gamma'$: // 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.



Proposed Framework: Formal Description and Overview



 $\mathcal{O} \leftarrow \emptyset$: // Conjectured streamliners $\Gamma \leftarrow \emptyset$: // Search streamliners // Search parameter $\rho \leftarrow \rho_0;$ $\mathcal{S} \leftarrow \emptyset$: // Solutions found $\tau \leftarrow false;$ // Timeout flag repeat $Solve(P_{\rho}, \Gamma, t) \rightarrow (S', \tau);$ // Search for new solutions if $\mathcal{S}' \cap \mathcal{S} \neq \emptyset$ then $S \leftarrow S \cup S'$: // Case 1: successful search Analyze(S) $\rightarrow O'$; // Conjecture new streamliners $\mathcal{O} \leftarrow \mathcal{O} \cup \mathcal{O}'$: $\rho \leftarrow \rho + 1$: else if τ is true then Select $\Gamma' \subset \mathcal{O}$; // Case 2: timed-out failed search $\Gamma \leftarrow \Gamma \cup \Gamma'$: // Strengthen streamliners else Select $\Gamma' \subset \Gamma$; // 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. 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**.















































e cait neip								_	Streamliner Editor
Solutions found		1-	1.	1	1.	1	1	_	Streamliner Name Symmetry
Streamliner \ Parameter	3	5	6	8	9	11	12		Evaluation Function
Any	12	5760	8736	238	411	9	6		for(int i=0; i <n; i++){<="" td=""></n;>
Reduced	1	2	14	12	1	1	0		if(a[]]]!=a[]]]){
Symmetry	6	240	8640	12	1	1	0		retum false;
Columns 2 and n	1	6	1	2	1	1	0		}' _
Cyclic	6	40	96	226	410	8	6		} retum true:
			_						Constraint Post Function
Selected Solutions	1 2	3 4 5	1 2	3 4 5	6 1	2 3	4 5 6	7 8	for(int i=0; i <n; for(int="" i="0;" i++){="" i++){<="" i<n;="" td=""></n;>
5:3194	2 4	5 3 1	2 4	6 5 3	1 2	4 6	8 7 5	3 1	cp.Add(cp.Eq(a[i][j],a[j][i]));
5:4526	3 5	2 1 4	3 6	4 1 2	5 3	6 8	5 2 1	4 7	}
5:5502	4 3	1 5 2	4 5	1 3 6	2 4	8 5	1 3 7	6 2	
5:5572 6:10	5 1	4 2 3	5 3	2 6 1	4 5	7 2	3 8 4	1 6	
8:10			6 1	5 2 4	3 6	5 1	7 4 2	8 3	
8:11					7	3 4	6 1 8	2 5	Cancel OK
					8	1 /	2 6 3	5 4	
I - Select Streamline	r Comh	ination		2 -	Set Par	ameter	. 1	- Perfo	orm search Search Stats
Reduced	Re	duced		Par	ameter n	11	т т	ime Limit	SO Search States
Symmetry	Sy Sy	nmetry			ametern				New solutions found
Columns 2 and n		iumns 2 and	n	Par	ameter k		🚊 C	lick to Rur	In Streamline Total # of solutions 9





🖳 Constructive Procedu File Edit Help	res Dis	covery T	ool					(Streamliner Editor
Solutions found									Straminer Name Summer
Streamliner \ Parameter	3	5	6	8	9	11	12	Π	Evaluation Function
Any	12	5760	8736	238	411	9	6		for(int i=0; i <n; i++){<="" td=""></n;>
Reduced	1	2	14	12	1	1	0		if(a[i])] != a[i]])){
Symmetry Columns 2 and n	1	240	8640	12 2	1	1	0		retum false; }
Cyclic	6	40	96	226	410	8	6		}
									retum true;
									Constraint Post Function
Selected Solutions 5:1464 5:3194 5:4526 5:4983 5:5502 5:5572 6:10 8:10 8:11	1 2 3 2 4 3 3 5 3 4 3 5 5 1 4	3 4 5 5 3 1 2 1 4 1 5 2 4 2 3	1 2 2 4 3 6 4 5 5 3 6 1	3 4 5 6 5 3 4 1 2 1 3 6 2 6 1 5 2 4	6 1 1 2 5 3 2 4 5 3 6 7 8	2 3 4 6 8 5 7 2 5 1 3 4 1 7	4 5 6 7 8 8 7 5 3 1 5 2 1 4 7 1 3 7 6 2 3 8 4 1 6 7 4 2 8 3 6 1 8 2 5 2 6 3 5 4		for(int i=0; i <n; i++){<br="">for(int j=0; j<n; j++){<br="">cp.Add(cp.Eq(a[i][j],a[j][i])); } } }</n;></n;>
1 - Select Streamliner Reduced Symmetry Columns 2 and n Cyclic	r Combi	nation duced ametry umns 2 and	n	2 - Para Para Crea Nan	Set Par ametern ameterk ate New St ne	ameter: 11 treamliner	s 3 - Pe		Search Search Stats





e Edit Help Solutions found								_	Streamliner Editor
Streamliner \ Parameter	3	5	6	8	9	11	12	_	Evaluation Evaluation
Any	12	5760	8736	238	411	9	6		for(int i=0; i <n; i++){<="" td=""></n;>
Reduced	1	2	14	12	1	1	0		for(int j=0; j <n; j++){<br="">#(⇒6001 U=⇒6000 M</n;>
Symmetry	6	240	8640	12	1	1	0		retum false;
Columns 2 and n	1	6	1	2	1	1	0		} ¹
Cyclic	6	40	96	226	410	8	6		}
								_	Constraint Post Function
Selected Solutions	1 2	3 4 5	1 2	3 4 5	6 1	2 3	4 5 6	7 8	for(int i=0; i <n; i++){<="" td=""></n;>
5:1464 5:3194	2 4	5 3 1	2 4	6 5 3	1 2	4 6	8 7 5	3 1	for(int j=0; j <n; j++){<br="">cp.Add(cp.Eg(a[i][i],a[i][i]));</n;>
5:4526	3 5	2 1 4	3 6	4 1 2	5 3	6 8	5 2 1	4 7	}
5:4983 5:5502	4 3	1 5 2	4 5	1 3 6	2 4	8 5	1 3 7	6 2	· · · · · · · · · · · · · · · · · · ·
5:5572	5 1	4 2 3	53	2 6 1	4 5	7 2	3 8 4	16	
Salaat atraat	nlin	\mathbf{h}	6 1	5 2 4	3 6	5 1	7 4 2	8 3	coarab
Sciect Sulear	111110	218, p	ai aii.		s, an	u pei	TOLI		Cancel OK
					8	7	2 6 3	5 4	
1 Select Streemlines	Comb	a official		-	Set Day			Deute	Samuel State
1 - Select Streaminer		duced		1 -	Set Par	ameters	; J	- Perio	rin search Search Stats
Symmetry	Syr	nmetry		Pan	ameter n	11	÷ T	ime Limit	bu 🔄 New solutions found 1
Columns 2 and n	Col	umns 2 and	n	Par	ameter k		÷	lick to Run	n Streamline Total # of solutions 9
0,0.0				Crea	ate New St	reamliner			for this order





ile Edit Select s	oluti	ons			1-	1			Streamliner Editor
Streamliner \ Parameter	3	5	6	8	9	11	12		Evaluation Function
Any Reduced	12	2	8/35	238	1	9	0		for(int i=0; i <n; i++){<br="">for(int j=0; j<n; j++){<="" td=""></n;></n;>
Symmetry	6	240	8640	12	1	1	0		if(a[i][i] != a[i][i]){ return false:
Columns 2 and n	1	6	1	2	1	1	0		}
Cyclic	6	40	96	226	410	8	6		}
									retum true;
Selected Solutions 5:1464 5:3194 5:4526 5:4983 5:5502 5:5572 6:10 8:10 8:11	1 2 3 2 4 5 3 5 2 4 3 1 5 1 4	4 5 3 1 2 1 4 5 2 2 4 2	1 2 2 4 3 6 4 5 5 3 6 1	3 4 5 6 5 3 4 1 2 1 3 6 2 6 1 5 2 4	6 1 1 2 5 3 2 4 4 5 3 6 7 8	2 3 4 6 8 8 5 7 2 5 5 1 7 3 4 9 1 7 2	4 5 6 7 8 7 5 3 5 2 1 4 1 3 7 6 3 8 4 1 7 4 2 8 6 1 8 2 2 6 3 5	8 1 7 2 6 3 5 4	<pre>for(int i=0; i<n; cancel="" cp.add(cp.eq(a[i][i],a[i][i]));="" for(int="" i++){="" j="0;" j++){="" j<n;="" ok<="" pre="" }=""></n;></pre>
1 - Select Streamliner Reduced Symmetry Columns 2 and n Cyclic	Combin Red Sym Colu	nation uced metry imns 2 and	n	2 - Par Par Cre Nar	Set Par ameter n ameter k ate New St ne	ameters 11 treamliner	3 - P ➡ Time L ➡ Click t	ertori imit o Run	60 Image: Search Stats Streamline New solutions found 1 Total # of solutions for this order 9 Total # of solutions 15172



le Edit Help Solutions found									Streamliner Editor
Streamliner \ Parameter	3	5	6	8	9	11	12		Evaluation Function
Any	12	5760	8736	238	411	9	6		for(int i=0: i <n; i++){<="" td=""></n;>
Reduced	1	2	14	12	1	1	0		for(int j=0; j <n; j++){<="" td=""></n;>
Symmetry	6	240	8640	12	1	1	0		retum false;
Columns 2 and n	1	6	1	2	1	1	0		}
Cyclic Analyze	solu	tions		226	410	8	6		}
	~ ~								retum true;
									Constraint Post Function
Selected Solutions	1 2 3	3 4 5	1 2 3	3 4 5	6 1	2 3	4 5 6	7 8	for(int i=0; i <n; i++){<="" td=""></n;>
5:1464	2 4 5	5 3 1	2 4 (6 5 3	1 2	4 6	8 7 5	3 1	for(int j=0; j <n; j++){<br="">cp.Add(cp.Eg(a[i][i].a[i][i]));</n;>
5:4526	3 5 2	2 1 4	3 6 4	4 1 2	5 3	6 8	5 2 1	4 7	}
5:4983 5:5502	4 3 1	52	4 5	1 3 6	2 4	8 5	1 3 7	6 2	3
5:5572	5 1 4	4 2 3	5 3 3	2 6 1	4 5	7 2	3 8 4	1 6	
8:10			6 1 4	5 2 4	3 6	5 1	7 4 2	8 3	
8:11					7	3 4	6 1 8	2 5	Cancel OK
					8	1 7	2 6 3	5 4	
1 Salast Stussed	Combi			-	Set Des		_		Security State
Peduced	Comor			2 -	set Par	ameters		5 - Perio	Search Stats
Symmetry	Sym	imetry		Para	ametern	11	-	Time Limit	bu New solutions found 1
Columns 2 and n	Colu	umns 2 and	n	Para	ameter k		×	Click to Run	n Streamline Total # of solutions 9
				Com	An Marco Co	Lange and Lange			tor this order





- 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





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

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





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







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 $a_{i,j} + i \le N$ and $j < N$ do
 $a_{i,j+1} = a_{i,j} + i;$
 $j = j + 1;$
else // Even sequence
while $a_{i,j-1} \ge 1$ 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
 $a_{i,j+1} = i - a_{i,j};$
 $k = k + 1;$
 $j = j + 1;$

ICS

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





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 $a_{i,j+1} = a_{i,j} + i;$
 $j = j + 1;$
else // Even sequence
while $a_{i,j+1} = a_{i,j} - i;$
 $j = j + 1;$
if $j < N$ then // Switch sequence
if k is odd then
 $a_{i,j+1} = 2N + 1 - i - a_{i,j};$
else
 $a_{i,j+1} = i - a_{i,j};$
 $k = k + 1;$
 $j = j + 1;$

ICS

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

Proof of Correctness in [R. Le Bras, A. Perrault, and C. Gomes, *Polynomial Time Construction for Spatially Balanced Latin Squares*, 2012]





ICS

Algorithm: SBLS-Cyclic procedure.





Application to 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.



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

Fig: Partition of [1,23] into 3 weakly sumfree sets, proving $WS(3) \ge 23$



Application to the Weak Schur problem



Best known lower bounds:

Approach	<i>WS</i> (5)	<i>WS</i> (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] (<i>revised</i>)



Application to the Weak Schur problem



Successful Key Streamliners:

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





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

Fig: Partition of [1,581] into 6 weakly sumfree sets, proving $WS(6) \ge 581$.

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





General **framework** that integrates specialized search techniques (so-called **streamlining**) with **human insight** in an **iterative** approach to discover **efficient constructive procedures**.

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

Improves the best known lower bound for the Weak Schur Number problem.

One exciting extension would be to **crowd-source** the search for regularities in the solution set.



	1	2 3	3 4	5	6 7	8	9	10	11 1	2 13	3 14	15	16	17	18 1	9 20	21	22	23	24 2	25 26	27	28	29	30	31 3	32 33	3 34	35 3	6 37	38	39 4) 41	42	43 4	4 45	46	47	48 49	50	NON
	2	4 6	5 8	10	12 1	4 16	18	20	22 2	24 20	6 28	30	32	34	36 3	8 40	42	44	46	48 5	0 49	47	45	43	41	39 3	37 35	5 33	31 2	9 27	25	23 2	1 19	17	15 1	3 11	9	7	5 3	1	See.
	3	6 9) 12	15	18 2	1 24	27	30	33 3	36 39	9 42	45	48	50	47 4	4 41	38	35	32	29 2	6 23	20	17	14	11	8	5 2	2 1	4	7 10	13	16 1	9 22	25	28 3	1 34	37	40	43 46	i 49	
	4	8 1	2 16	20	24 2	8 32	36	40	44 4	18 49	9 45	41	37	33	29 2	5 21	17	13	9	5	1 3	7	11	15	19	23 2	27 31	1 35	39 4	3 47	50	46 4	2 38	34	30 2	6 22	18	14	10 6	2	
	5	10 1	5 20	25	30 3	5 40	45	50	46 4	1 30	6 31	26	21	16	11 (5 1	4	9	14	19 2	4 29	34	39	44	49	47 4	42 37	7 32	27 2	2 17	12	7 2	3	8	13 1	8 23	28	33	38 43	48	
	6	12 1	8 24	30	36 4	2 48	47	41	35 2	29 23	3 17	11	5	1	7 1	3 19	25	31	37	43 4	9 46	40	34	28	22	16 1	10 4	2	8 1	4 20	26	32 3	3 44	50	45 3	9 33	27	21	15 9	3	
	7	14 2	1 28	35	42 4	9 45	38	31	24 1	7 10	0 3	4	11	18	25 3	2 39	46	48	41 :	34 2	27 20	13	6	1	8	15 2	22 29	9 36	43 5	0 44	37	30 2	3 16	9	2 5	; 12	19	26	33 40	47	A D:
	8	16 2	4 32	40	48 4	5 37	29	21	13	5 3	11	19	27	35	43 5	0 42	34	26	18	10 3	2 6	14	22	30	38	46 4	47 39	9 31	23 1	5 7	1	9 1	7 25	33	41 4	9 44	36	28	20 12	4	1900
	9	18 2	7 36	45	47 3	8 29	20	11	2	7 10	6 25	34	43	49	40 3	1 22	13	4	5	14 2	3 32	41	50	42	33	24	15 6	; 3	12 2	1 30	39	48 4	4 35	26	17 8	1	10	19	28 37	46	TTTT T
	10	20 3	0 40	50	41 3	1 21	11	1	9 1	9 2	9 39	49	42	32	22 1	2 2	8	18	28	38 4	8 43	33	23	13	3	7	17 27	7 37	47 4	4 34	24	14 4	6	16	26 3	6 46	45	35	25 15	5	NELL
	11	22 3	3 44	46	35 2	4 13	2	9	20 3	31 42	2 48	37	26	15	4 7	7 18	29	40	50 3	39 2	8 17	6	5	16	27	38 4	49 4	1 30	19	3 3	14	25 3	6 47	43	32 2	1 10	1	12	23 34	45	RSITY
	12	24 3	6 48	41	29 1	7 5	7	19	31 4	13 40	6 34	22	10	2	14 2	6 38	50	39	27	15	3 9	21	33	45	44	32 2	20 8	4	16 2	8 40	49	37 2	5 13	1	11 2	3 35	47	42	30 18	6	1
	13	26 3	9 49	36	23 1	1 3	16	29	42 4	16 31	3 20	7	6	19	32 4	5 43	30	17	4	9 2	2 35	48	40	27	14	1	12 2	5 38	50 3	7 24	11	2 1	5 28	41	47 3	4 21	8	5	18 31	44	1
	14	28 4	2 45	31	17 3	11	25	39	48 3	4 20	0 6	8	22	36	50 3	7 23	9	5	19	33 4	17 40	26	12	2	16	30 4	44 43	3 29	15	1 13	27	41 4	5 32	18	4 10	0 24	38	49	35 21	7	1
	15	30 4	5 41	26	11 4	19	34	49	37 2	2 7	8	23	38	48	33 1	8 3	12	27	42	14 2	9 14	1	16	31	46	40 2	25 10	0 5	20 3	5 50	36	21 6	9	24	39 4	7 32	17	2	13 28	43	1
	16	32 4	8 37	21	5 1	1 27	43	42	26 1	0 6	22	38	47	31	15 1	1 17	33	49	36	20 4	4 12	28	44	41	25	9	7 2	3 39	46 3	0 14	2	18 3	4 50	35	19 3	13	29	45	40 24	8	1
	17	34 5	0 33	16	1 1	8 35	49	32	15	2 19	9 36	48	31	14	3 2	0 37	47	30	13	4 2	1 38	46	29	12	5	22 3	39 44	5 28	11 (5 23	40	44 2	7 10	7	24 4	1 43	26	9	8 25	42	1
	18	36 4	7 29	11	7 2	5 43	40	22	4 1	4 32	2 50	33	15	3	21 3	9 44	26	8	10	28 4	6 37	19	1	17	35	48 3	30 12	2 6	24 4	2 41	23	5 1	3 31	49	34 10	6 2	20	38	45 27	9	1
	19	38 4	4 25	6	13 3	2 50	31	12	7 2	6 4	5 37	18	1	20	39 4	3 24	5	14	33	49 3	0 11	8	27	46	36	17	2 2	1 40	42 2	3 4	15	34 4	3 29	10	9 2	8 47	35	16	3 22	41	
	20	40 4	1 21	1	19 3	9 42	22	2	18 3	18 41	3 23	3	17	37	44 2	4 4	16	36	45 3	25 1	5 15	35	46	26	6	14 3	34 47	7 27	7 1	3 33	48	28 8	12	32	49 2	9 9	11	31	50 30	10	
	21	42 3	8 17	4	25 4	6 <u>34</u>	13	8	29 5	50 30	0 9	12	33	47	26	5 16	37	43	22	1 2	0 41	39	18	3	24	45 3	35 14	4 7	28 4	9 31	10	11 3	2 48	27	6 1	5 36	44	23	2 19	40	
	22	44 3	5 13	9	31 4	8 26	4	18	40 3	129 17	7 5	27	49	30	8 1	4 36	43	21	1 :	23 4	5 34	12	10	32	47 3	25	3 19	9 41	38 1	6 6	28	50 2	3 7	15	37 4	2 20	2	24	46 33	11	
	23	46 3	2 9	14	37 4	1 18	5	28	50 2	7 4	19	42	36	13	10 3	3 45	22	1	24	17 3	1 8	15	38	40	17	6 2	29 49	9 26	3 2	0 43	35	12 1	1 34	44	21 2	25	48	30	7 16	39	
	24	48 2	95	19	43 3	4 10	14	38	39 1	5 9	33	44	20	4	28 4	9 25	1	23	47	30 0	6 18	42	35	11	13	37 4	40 16	6 8	32 4	5 21	3	27 5	1 26	2	22 4	6 31	7	17	41 36	12	
	25	50 2	6 1	24	49 2	7 2	23	48	28	3 2	2 47	29	4	21	46 3	0 5	20	45	31	6 1	9 44	32	7	18	43	33	8 17	7 42	34 9	9 16	41	35 1	1 15	40	36 1	1 14	39	37	12 13	38	
	26	49 2	33	29	46 2	0 6	32	43	17	9 3	5 40	14	12	38	37 1	1 15	41	34	8	18 4	4 31	5	21	47	28	2 3	24 50	0 25	1 2	7 48	22	4 3	1 45	19	7 3	3 42	16	10	36 39	13	
	27	47 2	0 7	34	40 1	3 14	41	33	6 2	21 45	8 26	1	28	46	19 3	2 25	39	12	15	12 3	2 5	22	49	25	2	29 4	15 15	8 9	36 3	8 11	16	43 3	1 4	23	50 2	4 3	30	44	17 10	37	
	28	45 1	7 11	39	34 6	22	50	23	5 3	33 40	1 12	16	44	29	1 2	7 46	18	10	38	35	7 21	49	24	4	32	41 1	13 19	5 43	30 3	2 26	47	19 9	37	36	8 2	0 48	25	3	31 42	14	
	29	43 1	4 15	44	28 1	30	42	13	16 4	15 2	7 2	31	41	12	17 4	6 26	3	32	40	11 1	8 47	25	4	33	39	10 1	19 45	8 24	5 3	4 38	9	20 4	3 23	6	25 2	7 8	21	50	22 7	36	
	30	41 1	1 19	49	22 8	38	33	3	27 4	14 14	4 16	46	25	5	35 3	a a	24	47	17	13 4	3 28	2	32	39	9	21 4	50 20	0 10	40 3	1 1	29	42 1	2 18	48	23 7	37	34	4	26 45	15	1
6	31	39 8	23	47	16 1	5 46	24	7	38 3	2 1	30	40	9	22	48 1	7 14	45	25	6 :	37 3	3 2	29	41	10	21	49 1	18 13	3 44	26	5 36	34	3 2	3 42	11	20 5	0 19	12	43	27 4	35	
	32	37 5	i 27	42	10 2	2 47	15	17	49 2	20 12	2 44	25	7	39	30 3	2 34	35	3	29	10 1	8 24	45	13	19	50	18 1	14 46	6 23	9 4	1 28	4	36 3	3 1	31	38 6	26	43	11	21 48	16	
	33	35 2	31	37	4 2	9 39	6	27	41	8 2	5 43	10	23	45	12 2	1 47	14	19	49	16 1	7 50	18	15	48	20	13 4	46 23	2 11	44 2	4 9	42	26 7	40	28	5 3	8 30	3	36	32 1	34	
	34	33 1	35	32	2 3	6 31	3	37	30	4 3	8 29	5	39	28	6 4	0 27	7	41	26	8 4	2 25	9	43	24	10 4	44 2	23 11	1 45	22 1	2 46	21	13 4	7 20	14	48 19	9 15	49	18	16 50	17	
-63	35	31 4	1 39	27	8 4	3 23	12	47	19 1	6 50	0 15	20	46	11	24 4	2 7	28	38	3 :	32 3	4 1	36	30	5	40	26	9 44	4 22	13 4	8 18	17	49 1	4 21	45	10 2	5 41	6	29	37 2	33	
\mathbf{z}	36	29 7	7 43	22	14 5	0 15	21	44	8 2	28 3	7 1	35	30	6	42 2	3 13	49	16	20	45 9	9 27	38	2	34	31	5 4	41 24	4 12	48 1	7 19	46	10 2	5 39	3	33 3	2 4	40	25	11 47	18	
Ľ	37	27 1	0 47	17	20 4	4 7	30	34	3 4	10 24	4 13	50	14	23	41 4	1 33	31	6	43	21 1	6 48	11	26	38	1 :	36 2	28 9	46	18 1	9 45	8	29 3	5 2	39	25 12	2 49	15	22	42 5	32	
	38	25 1	3 50	12	26 3	7 1	39	24	14 4	19 11	1 27	36	2	40	23 1	5 48	10	28	35	3 4	1 22	16	47	9	29	34	4 42	2 21	17 4	6 8	30	33 5	43	20	18 4	5 7	31	32	6 44	19	
	39	23 1	6 46	7	32 3	0 9	48	14	25 3	37 2	41	21	18	44	5 3	4 28	11	50	12	27 3	5 4	43	19	20	42	3 3	36 20	6 13	49 1	0 29	33	6 4	5 17	22	40 1	38	24	15	47 8	31	
	40	21 1	9 42	2	38 2	3 17	44	4	36 2	25 1	5 46	6	34	27	13 4	8 8	32	29	11 !	50 1	0 30	31	9	49	12	28 3	33 7	47	14 2	6 35	5	45 1	5 24	37	3 4	3 18	22	39	1 41	20	
	41	19 2	2 38	3	44 10	6 25	35	6	47 1	3 2	8 32	9	50	10	31 2	9 12	48	7	34	26 1	5 45	4	37	23	18	42	1 40	0 20	21 3	9 2	43	17 2	4 36	5	46 14	4 27	33	8	49 11	30	
	42	17 2	5 34	8	50 9	33	26	16	43	1 4	1 18	24	35	7	49 1	0 32	27	15	44	2 4	0 19	23	36	6	48	11 3	31 28	8 14	45 3	3 39	20	22 3	7 5	47	12 3	0 29	13	46	4 38	21	
	43	15 2	8 30	13	45 2	41	17	26	32 1	1 4	7 4	39	19	24	34 9	49	6	37	21	22 3	6 7	50	8	35	23	20 3	38 5	48	10 3	3 25	18	40 3	46	12	31 2	7 16	42	1	44 14	29	
	44	13 3	1 26	18	39 5	49	8	36	21 2	23 34	4 10	47	3	41	16 2	8 29	15	42	2	16 1	1 33	24	20	37	7	50	6 39	8 19	25 3	2 12	45	1 4	3 14	30	27 1	7 40	4	48	9 35	22	1
	45	11 3	4 22	23	33 1	2 44	1	46	10 3	35 2	1 24	32	13	43	2 4	7 9	36	20	25	31 1	4 42	3	48	8	37	19 2	26 30	0 15	41 4	4 49	7	38 1	3 27	29	16 4	0 5	50	6	39 17	28	
	46	9 3	7 19	28	27 1	36	10	45	1 4	17 8	38	17	29	26	20 3	5 11	44	2	48	7 3	9 16	30	25	21	34	12	43 3	49	6 4	0 15	31	24 2	2 33	13	42 4	50	5	41	14 32	23	
	47	7 4	0 14	33	21 2	5 28	19	35	12 4	2 5	49	2	45	9	38 1	6 31	23	24	30	17 3	7 10	44	3	50	4	43 1	11 30	6 18	29 2	5 22	32	15 3	8 6	46	1 4	8 6	41	13	34 20	27	
	48	5 4	3 10	38	15 3	3 20	28	25	23	30 11	8 35	13	40	8	45	3 50	2	46	7 4	41 1	2 36	17	31	22	26	27	21 3	2 16	37 1	1 42	6	47 1	49	4	44 9	39	14	34	19 29	24	44
	49	3 4	6 6	43	9 4	12	37	15	34 1	8 3	1 21	28	24	25	27 2	2 30	19	33	16	36 1	3 39	10	42	7	45	4 4	48 1	50	2 4	7 5	44	8 4	1 11	38	14 3	5 17	32	20	29 23	26	
	50	1 4	9 2	48	3 4	7 4	46	5	45	6 44	4 7	43	8	42	9 4	1 10	40	11	39	12 3	8 13	37	14	36	15	35	16 34	4 17	33 1	8 32	19	31 2	3 30	21	29 2	2 28	23	27	24 26	25	
	-		_				-			11	1	1		-1			1		1			1		- 1							-	-					1				l

Extra slides





SBLS of order 50 – Construction 1



	-			-	- 1	- 1	- 1	- 1																																					1			
1	2	3	4	5	6	7	8	9 1	10 1	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 1	18 2	20 2	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 2	27 3	30 3	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 3	36 4	40 4	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 4	15 5	50 4	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 4	17 4	41 3	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 3	38 3	31 2	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 2	29 2	21 1	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 2	20 1	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 1	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 2	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 1	19 3	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 1	16 2	29 4	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 2	25 3	39 4	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 3	34 4	49 3	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 4	13 4	42 2	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 4	19 3	32 1	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 4	10 2	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 3	1 1	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 2	22	2 1	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 1	13	8 2	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 1	18 4	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
22	44	32	9	14	37	41	18	5 2	28 5	50	27	4	19	12	26	12	10	22	45	22	1	24	47	21	0	15	20	40	17	6	29	19	26	2	20	13	25	12	11	24	44	21	2	25	49	20	7	16 29
24	40	29	5	19	12	24	10 1		20 1	29	15	9	22	44	20	4	29	19	25	1	22	47	20	6	19	12	25	11	12	27	40	16	20	22	45	21	2	27	50	26	2	22	46	21	7	17	41	26 17
24	50	25	1	24	40	27	2 2	2	19 1	20	2	22	47	29	20	21	46	20	5	20	45	21	6	19	10	32	7	19	13	22	-+0	17	12	34	4J Q	16	41	25	10	15	40	26	11	14	20	27	12	12 29
20	49	22	2	29	45	20	6 3	22	12 1	17	9	25	40	14	12	20	27	11	15	41	24	0	19	10	21	5	21	17	29	2	24	50	25	1	27	10	22	4	20	45	19	7	22	17	16	10	26	20 12
20	43	20	7	24	40	12	14 4	11 3	+J 22	6	21	40	20	1	20	10	10	0	25	20	12	15	42	22	51	22	40	25	20	20	45	10	0	20	20	11	10	42	21	45	22	,	24	-42	20	44	17	10 27
20	47	17	11	20	24	6	22 6	0 1	22	5	21	40	12	16	20	20	1.3	27	46	10	10	20	92	7	21	40	24	2.5	2	41	12	16	42	20	20	20	47	10	31	- 4 27	20	0	24	40	26	2	21	42 14
20	40	14	15	33	24	1	22 0	12 1	12 1	0 10	33	40	12	21	44	12	17	21	40	2	22	30	11	10	47	45	24	4	20	10	10	10	43	50	2	20	4/	20	3	22	20	0	20	40	20	50	22	7 20
20	43	14	10	44	20	-	30 4	12	1.3	10	45	14	10	31	41	12	17	40	20	3	32	40	12	10	4/	20	4	20	33	10	13	40	10	5	34	30	3	20	43	10	0	30	3/	0	21	50	22	/ 30
30	41	0	19	49	10	0 15	38 3	53	3 4	2/	44	14	16	46	25	2	30	30	5	24	4/	0	13	43	28	2	32	39	3	21	10	20	10	40	51	1	29	42	12	18	48	23	/	3/	34	4	20	40 10
31	33	0 E	23	47	10	10	40 2	24 LE 1	17	40	32	12	30	40	3	22	40	2	14	40	20	20	3/	33	2	23	41	10	21	43	10	1.5	44	20	3	20	34	200	20	42	11	20	00	13	12	43	21	4 30
32	37	2	21	42	10	22	4/ 1		27 4	49	20	12	44	20	22	39	30	2	34	30	3	29	40	8	24	40	13	19	20	18	14	46	23	9	41	28	4	36	33	1	31	58	20	20	43	20	21	48 16
33	30	2	31	37	4	29	39	04	27 4	41	8	25	43	10	23	40	12	21	47	14	19	49	16	17	00	18	15	48	20	13	46	22	11	44	24	9	42	26	/	40	28	5	38	30	3	36	32	1 34
34	33	+	35	32	2	36	31	3 3	3/ 3	30	4	38	29	5	39	28	6	40	2/	/	41	26	8	42	25	9	43	24	10	44	23		45	22	12	46	17	13	4/	20	14	48	19	15	49	18	16	50 17
30	31	4	39	2/	8	43	23 1	2 4	4/	19	16	00	15	20	46		24	42	/	28	38	3	32	34	07	36	30	2	40	26	9	44	22	13	48	18	17	49	14	21	40	10	20	41	6	29	3/	2 33
36	29	-	43	22	14	00	10 4		44	8	28	3/	10	35	30	6	42	23	13	49	16	20	45	9	2/	38	2	34	31	2	41	24	12	48	17	19	46	10	26	39	3	33	32	4	40	20	11	4/ 18
3/	2/	10	4/	1/	20	44	/ 3	30 3	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 3	39 2	24	14	49	11	2/	36	2	40	23	15	48	10	28	35	3	41	22	16	4/	9	29	34	4	42	21	1/	46	8	30	33	5	43	20	18	45	/	31	32	6	44 19
39	23	16	46	/	32	30	9 4	18	14 2	25	37	2	41	21	18	44	5	34	28	11	50	12	2/	35	4	43	19	20	42	3	36	26	13	49	10	29	33	6	45	17	22	40	1	38	24	15	4/	8 31
40	21	19	42	2	38	23	17 4	14	4 3	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 3	35	6 4	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 2	26 1	16 4	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 1	17 2	26 3	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 3	36 2	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 4	46 1	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 1	10 4	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 1	19 3	35 1	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 2	28 2	25 2	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 3	37 1	15 3	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 4	16	5 4	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



Results of SBLS-sequence (U&B)



Order	CP	CPSS	CPCS	LSS	U&B
6	0.06	0.05	0.02	0.00	0.00
8	16.00	0.88		0.00	0.00
9	241.00	0.91		0.00	0.00
11		9.84		0.00	0.00
12		531.00	14.40	0.00	
14		$5,\!434.00$		0.02	0.00
15				0.01	0.00
17				0.25	
18			107,000.00	2.30	0.00
20				16.00	0.00
21				16.00	0.00
23				104.00	0.00
24				281.00	
26				609.00	0.00
27				4,000.00	
29				23,000.00	0.00
30				160,000.00	0.00
32				1,200,000.00	
33				1,200,000.00	0.00
35				1,200,000.00	0.00
36					0.00
39					0.00
41					0.00
44					0.00
48					0.00
50					0.00
51					0.00
53					0.00
54					0.00
56					0.00
999					0.02

