Using BOINC to enumerate mutually orthogonal Latin squares

Gerdus Benadé, Alewyn Burger, Jan van Vuuren

Grenoble, 25 Spetember 2013

Latin squares

Definition (Latin square)

A Latin square of order n is an $n \times n$ square where every cell contains one of the symbols in the set $\{0, 1, \ldots, n-1\}$ such that no symbol is repeated in any row or column.

2	1	0	3
0	3	2	1
3	0	1	2
1	2	3	0

2	0	3	1
1	3	0	2
0	2	1	3
3	1	2	0

Latin squares - Orthogonality

Definition

Two latin squares of the same order are *orthogonal* if, when superimposed, each of the possible n^2 ordered pairs occur exactly once.

$$\begin{array}{cccc} 2 & 1 & 0 & 3 \\ 0 & 3 & 2 & 1 \\ 3 & 0 & 1 & 2 \\ 1 & 2 & 3 & 0 \end{array} \qquad \begin{array}{cccc} 2 & 0 & 3 & 1 \\ 1 & 3 & 0 & 2 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 2 & 0 \end{array}$$

$$\begin{array}{cccc} 2,2 & 1,0 & 0,3 & 3,1 \\ 0,1 & 3,3 & 2,0 & 1,2 \\ 3,0 & 0,2 & 1,1 & 2,3 \\ 1,3 & 2,1 & 3,2 & 0,0 \end{array}$$

Latin squares - Orthogonality

Definition

A set $\{L_1, \ldots, L_k\}$ of $k \geq 2$ latin squares of order n is *orthogonal* if any two distinct latin squares are orthogonal. We call this a set of k *mututally orthogonal latin squares* (k-MOLS) of order n.

$$\begin{array}{cccc} 2 & 1 & 0 & 3 \\ 0 & 3 & 2 & 1 \\ 3 & 0 & 1 & 2 \\ 1 & 2 & 3 & 0 \end{array} \qquad \begin{array}{cccc} 2 & 0 & 3 & 1 \\ 1 & 3 & 0 & 2 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 2 & 0 \end{array} \qquad \begin{array}{cccc} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{array}$$

Permutations and main classes

Permuting/re-arranging the rows, symbols and columns of a MOLS give a structurally similar MOLS.

Definition (Main class)

A MOLS, together with all its permutations, is called a *main class.*

Definition (Class representative)

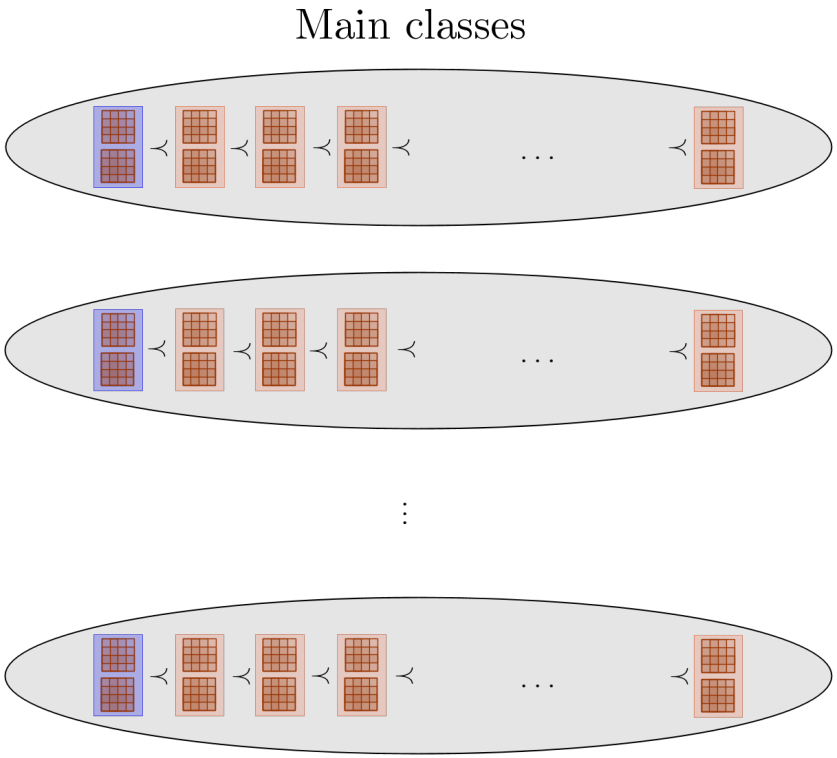
Every main class has a lexicographical smallest element, called the *class representative*.

Enumerating main classes of MOLS

Problem

How many main classes of k-MOLS of order n are there?

Approach: Count the class representatives



Enumerating main classes of MOLS

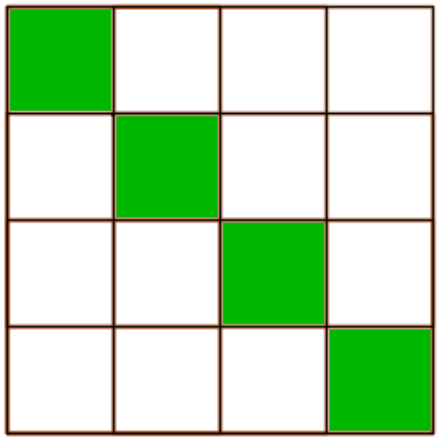
Problem

How many main classes of k-MOLS of order n are there?

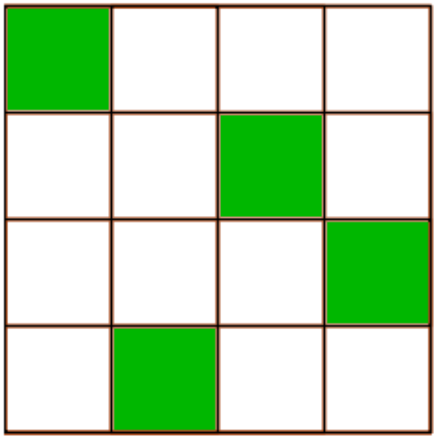
n	k							
	2	3	4	5	6	7	8	9
3	1							
4	1	1						
5	1	1	1					
6	0	0	0	0				
7	7	1	1	1	1			
8	2 165	39	1	1	1	1		
9	≥ 1	≥ 1	≥ 1	≥ 1	≥ 1	≥ 1	19	
10	≥ 1	?	?	?	?	0	0	0

Table: The number of structurally different k-MOLS of order n for $n \in \{3, 4, \ldots, 10\}$.

Enumeration algorithm - Universals

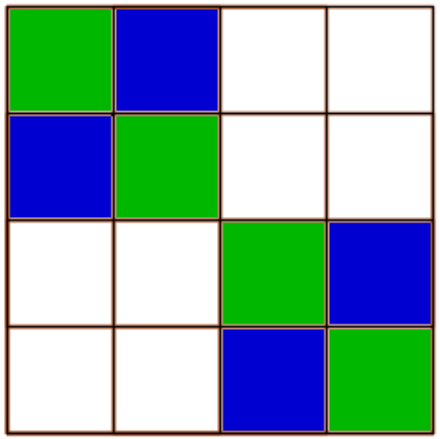


$< 0, 1, 2, 3 >$

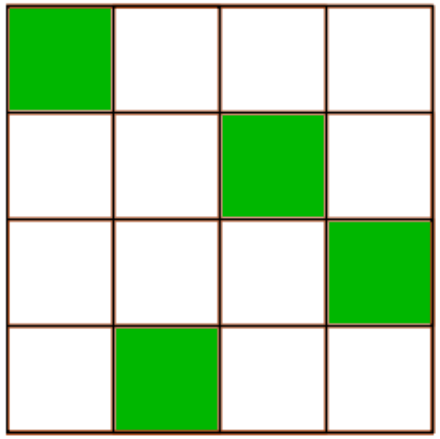


$< 0, 2, 3, 1 >$

Enumeration algorithm - Universals



$< 0, 1, 2, 3 >$
$< 1, 0, 3, 2 >$

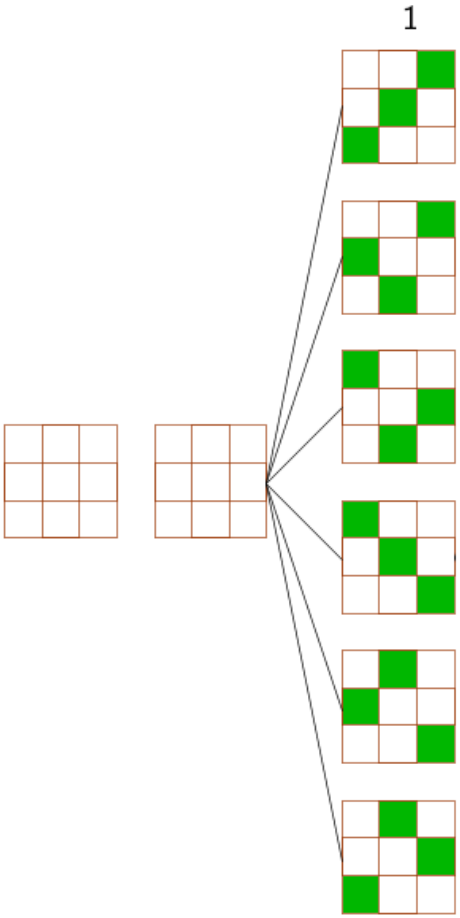


$< 0, 2, 3, 1 >$

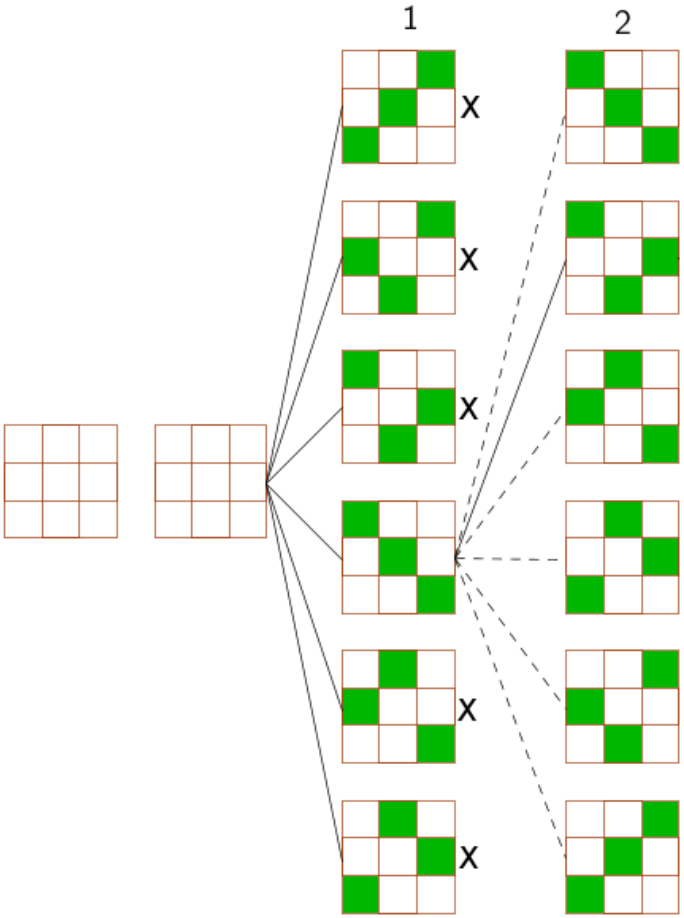
A partial MOLS needs certain properties.

$$\mathbb{P} = \begin{cases} \text{Every square in } M \text{ has to be 'Latin'} \\ \text{All pairs in } M \text{ orthogonal} \\ \text{No 'smaller' partial MOLS in this main class} \end{cases}$$

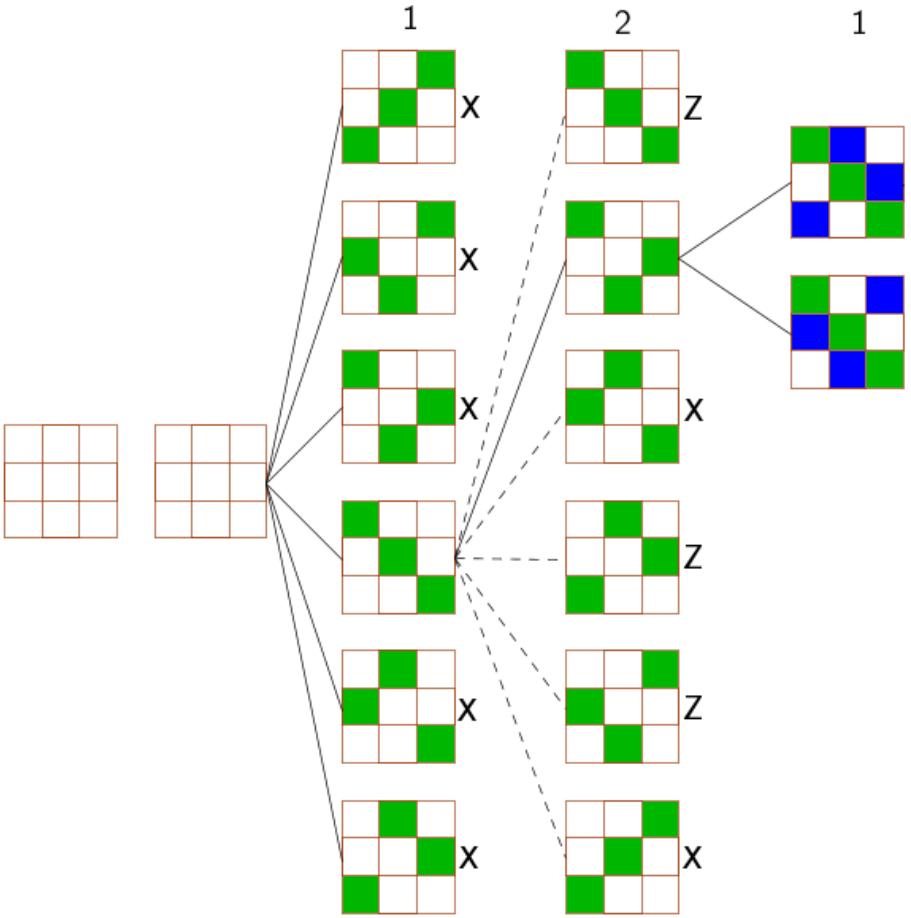
Enumeration algorithm - Example



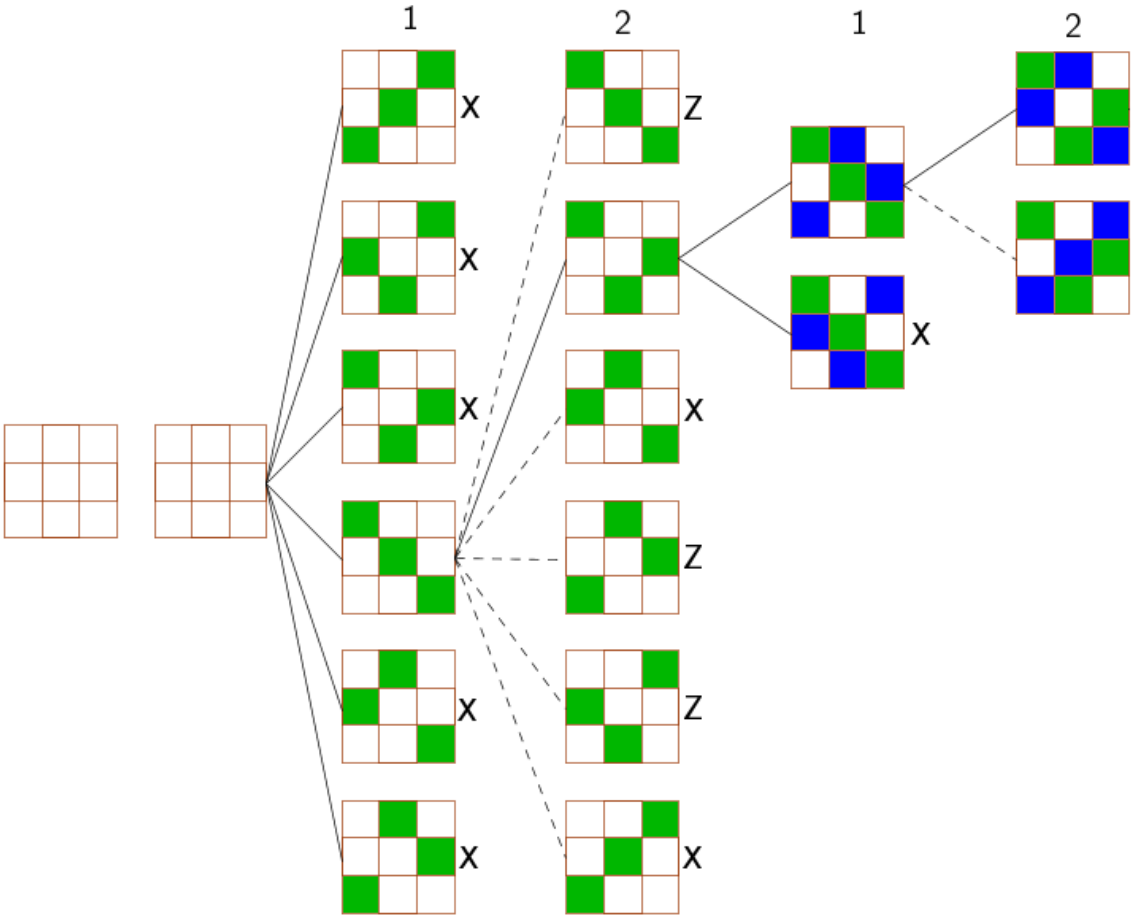
Enumeration algorithm - Example



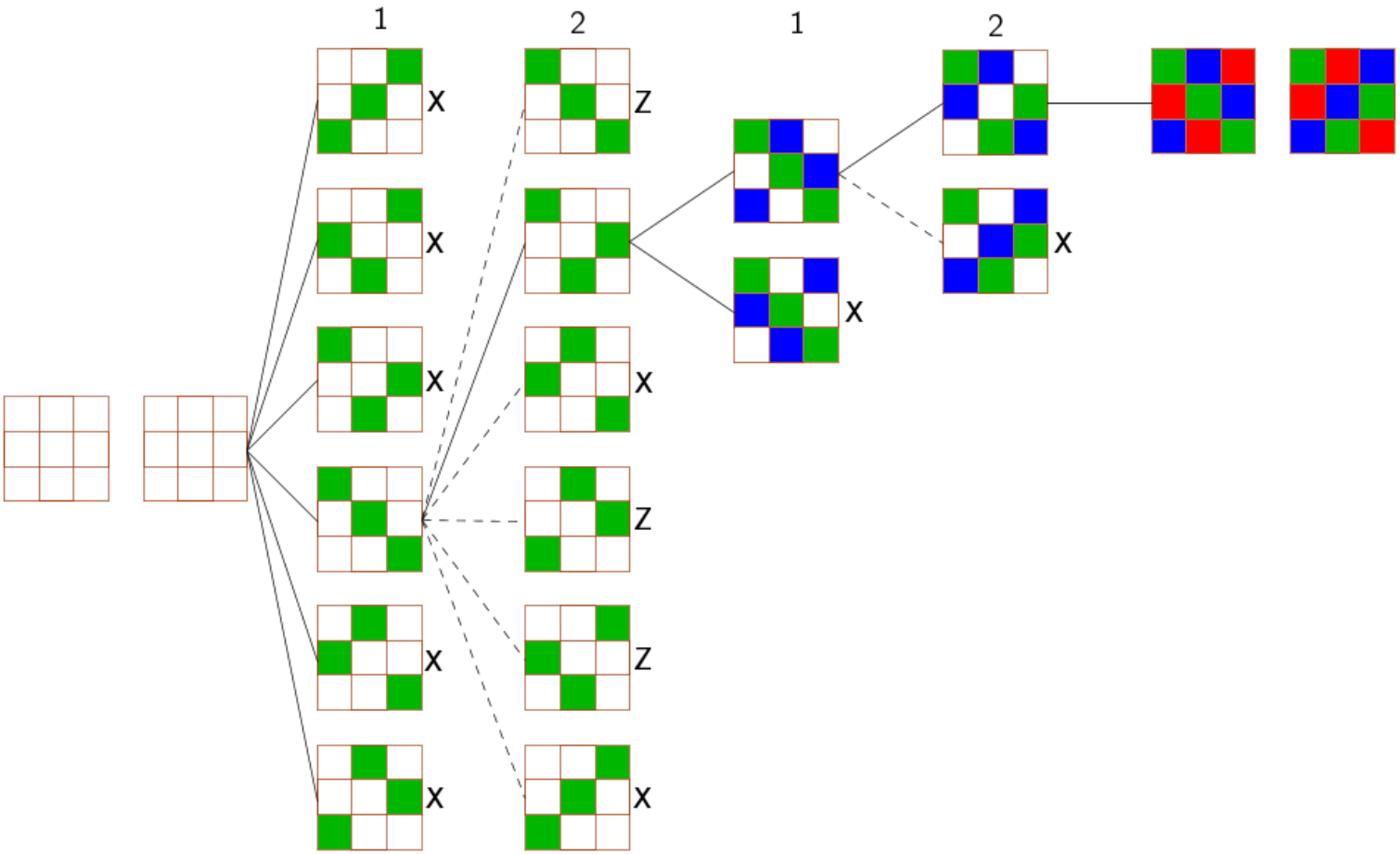
Enumeration algorithm - Example



Enumeration algorithm - Example



Enumeration algorithm - Example



Enumeration algorithm - Branches

Table: The number of branches on after every symbol for a 3-MOLS of order n.

	After universal					
n	1	2	3	4	5	6
3	1	1	1			
4	1	1	1	1		
5	2	4	2	2	1	
6	3	20	0	0	0	0
7	14	10529	3800	3	3	3
8	45	15 948 763	1 546 241 258	18 877 734	216	168
9	269	2.89×10^{10}	8.48×10^{14}	2.68×10^{15}		
10	1700	1.21×10^{14}	2.42×10^{21}	—		

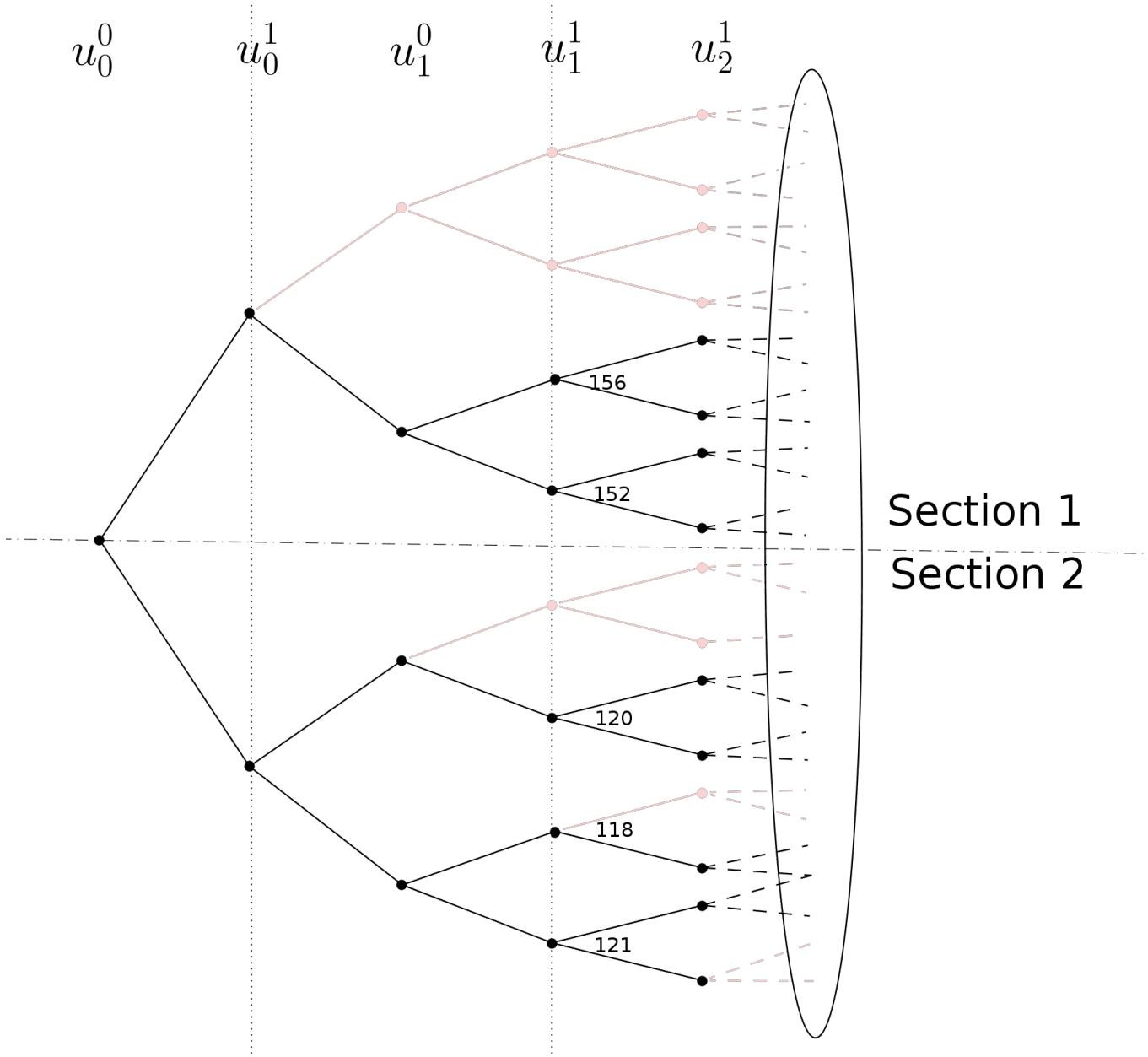
The number of nodes in the tree grows very quickly, making the complete enumeration for orders 9 and up difficult, if not impossible.

What next?

n	k							
	2	3	4	5	6	7	8	9
3	0							
4	0	0						
5	0	0	0					
6	0	0	0	0				
7	4	6	2	1	1			
8	$8d$	$9d$	$3h$	$1h$	$650s$	$113s$		
9		$\approx 14y$						
10		$\approx 60y$						

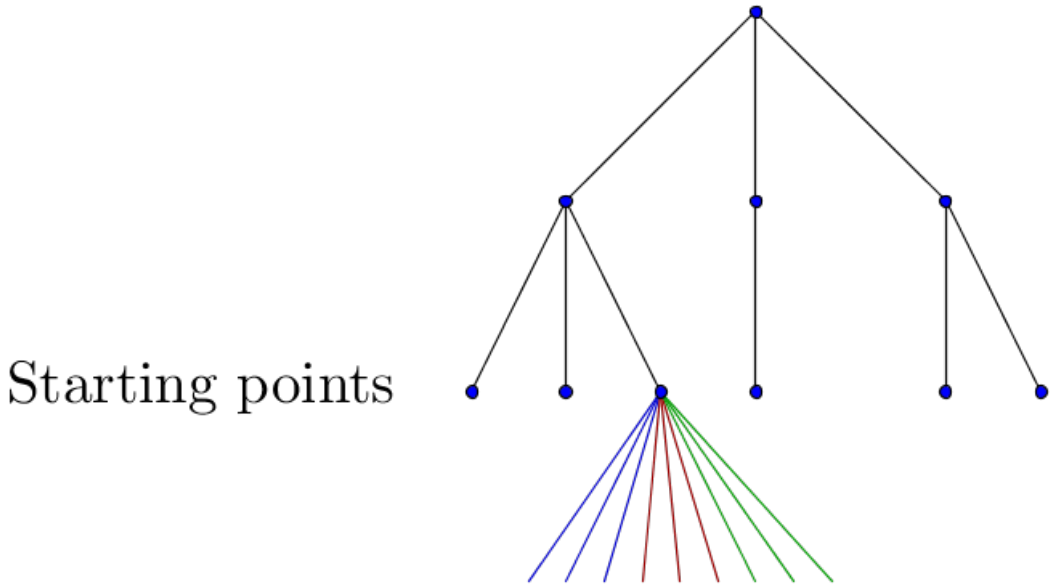
- Fix graphics, portability, GPU
- Test within our university network
- Public launch if feasible

Natural partition

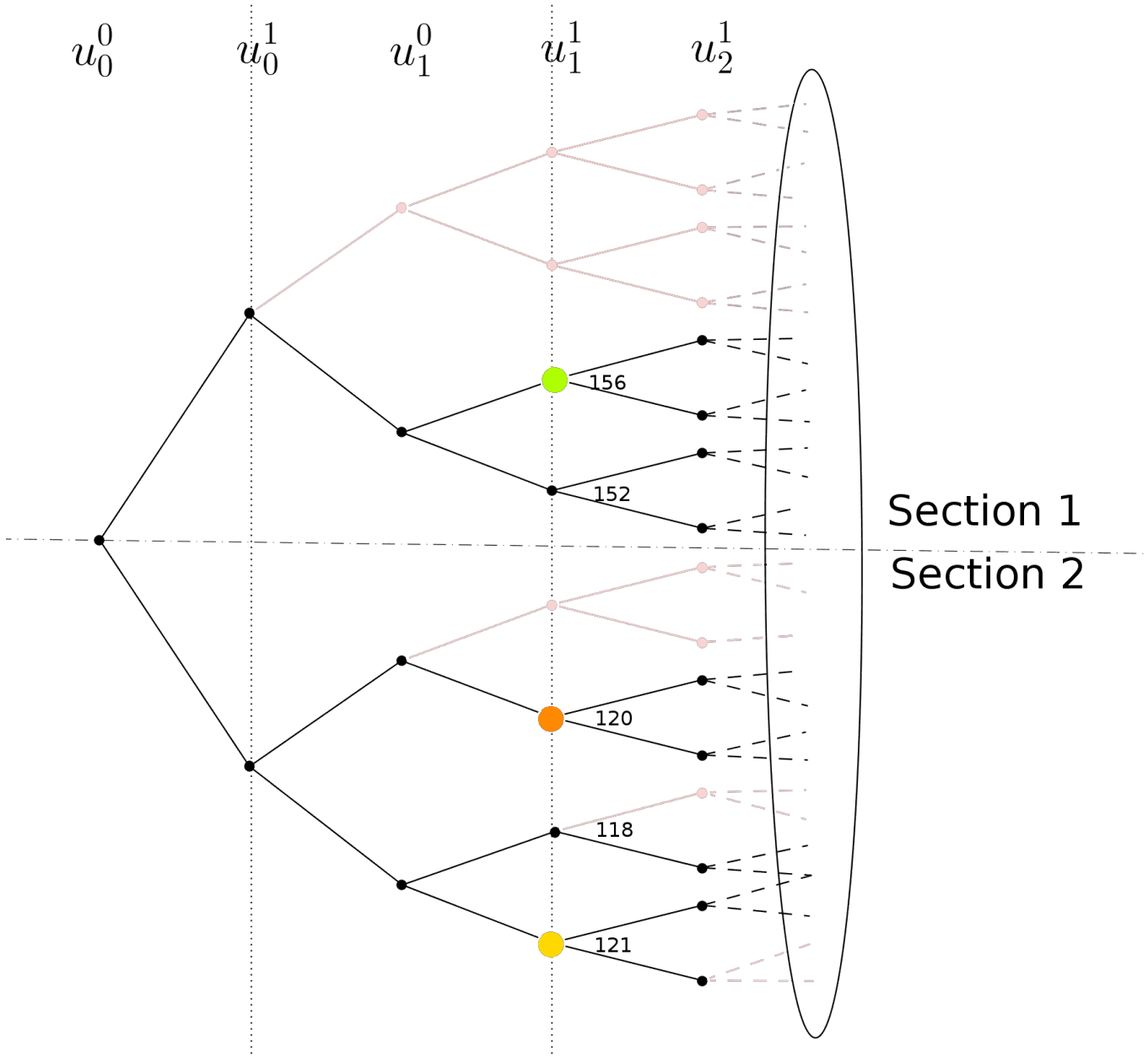


Job size estimation

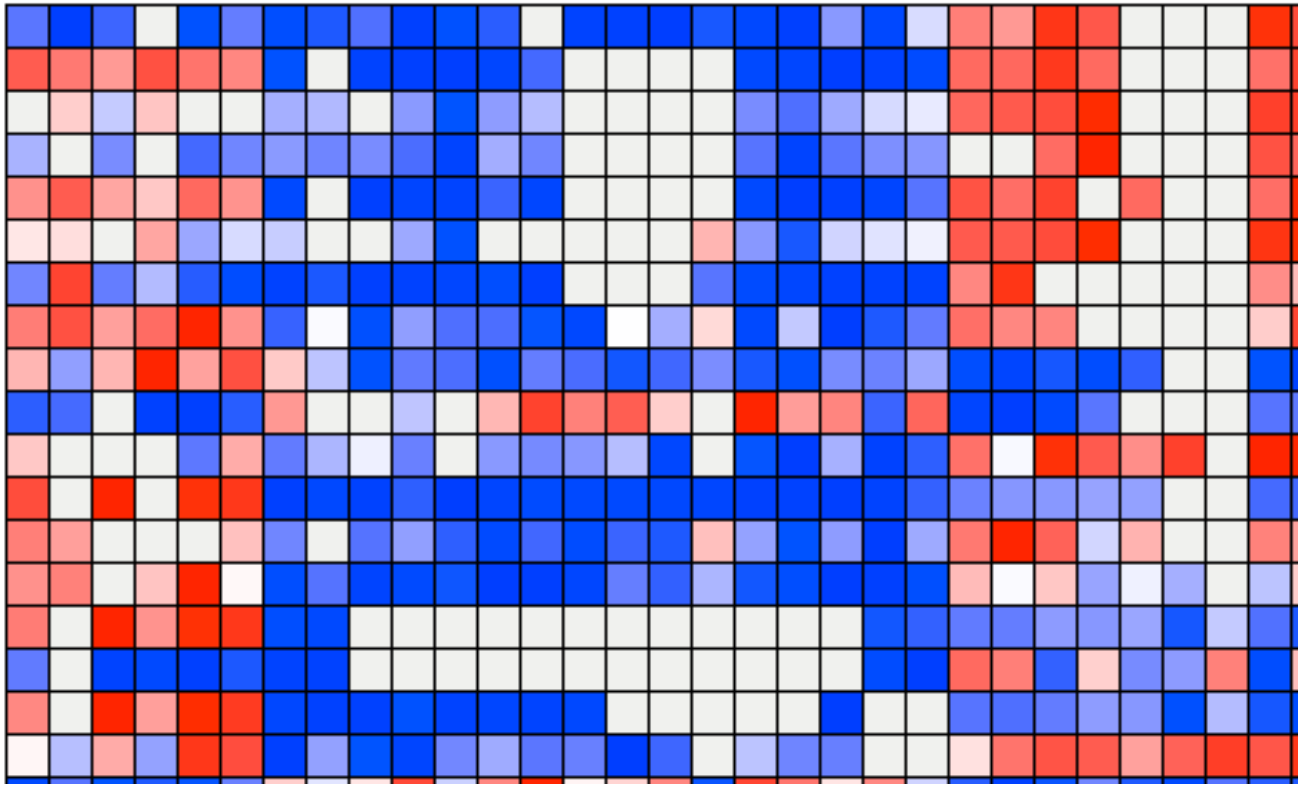
- Job size range from 1 second to 90 000 (3-MOLS order 8) – hard to provide accurate estimate and maximum fpops
- Workunits are too long, but there are too many to generate them on a deeper level. Enumerate partial subtrees using checkpoints as new workunits.



Interactivity



Interactivity



Keep very little work on hand, use visualization to generate workunits on demand (suitable?).
Alternative: "Poll" to decide which area to explore next.
Should work best with shorter workunits, see the consequences of your choices