Invariants for efficiently computing the autotopism group of a partial Latin rectangle

Daniel Kotlar

Tel-Hai College, Israel

joint work with

Eiran Danan, Raúl M. Falcón, Trent G. Marbach, Rebecca J. Stones

Symmetry vs Regularity, Pilsen, 2018

Partial Latin Rectangle

Partial Latin Rectangle

$$L \in \mathrm{PLR}(r, s, n)$$

$r =$ number of rows

$s =$ number of columns

$n =$ number of symbols

Partial Latin Rectangle

$$L \in \mathrm{PLR}(r, s, n)$$

$r =$ number of rows

$s =$ number of columns

$n =$ number of symbols

1	·	2	·	·	·	3	·	·
2	·	·	4	1	5	6	·	
·	1	5	3	·	4	·	·	·
·	2	·	5	·	3	·	4	·
4	3	·	·	5	·	1	·	2
·	·	·	·	2	·	·	1	3

$$\in \mathrm{PLR}(6, 9, 7)$$

Partial Latin Rectangle

$$L \in \mathrm{PLR}(r, s, n)$$

$r =$ number of rows

$s =$ number of columns

$n =$ number of symbols

1	·	2	·	·	·	3	·	·
2	·	·	4	1	5	6	·	
·	1	5	3	·	4	·	·	·
·	2	·	5	·	3	·	4	·
4	3	·	·	5	·	1	·	2
·	·	·	·	2	·	·	1	3

$$\in \mathrm{PLR}(6, 9, 7)$$

The entry set of L

$$\mathrm{Ent}(L) := \{(i, j, L[i, j]) : i \in [r], j \in [s], L[i, j] \in [n]\}$$

Isotopisms and autotopisms

$\Theta = (\alpha, \beta, \gamma) \in S_r \times S_s \times S_n$ an isotopism

$\Theta : \mathrm{PLR}(r, s, n) \rightarrow \mathrm{PLR}(r, s, n)$

α permutes the rows

β permutes the columns

γ permutes the symbols

If $\Theta(L) = L$ then Θ is an autotopism of L

$\mathrm{Atop}(L) =$ The autotopy group of L

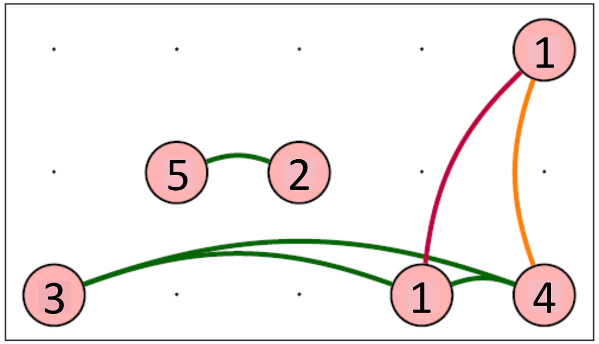
Computing $\mathrm{Atop}(L)$

The partial Latin rectangle graph

(Falcon and Stones, Disc. Math. 2017)

Example:

·	·	·	·	1
·	5	2	·	·
3	·	·	1	4



Compute $\mathrm{Atop}(L)$ by computing graph automorphisms.

Use ‘nauty’ (McKay, Meynert, Myrvold, JCD 2007) and its variants

Two stages in Computing $\mathrm{Atop}(L)$

1. Use invariants to partition rows, columns, symbols and entries in order to narrow down the search - polynomial time complexity.

Two stages in Computing $\mathrm{Atop}(L)$

1. Use invariants to partition rows, columns, symbols and entries in order to narrow down the search - polynomial time complexity.
2. Compute $\mathrm{Atop}(L)$ using backtracking methods - mostly non-polynomial time complexity.

Two stages in Computing $\mathrm{Atop}(L)$

1. **Use invariants to partition rows, columns, symbols and entries in order to narrow down the search - polynomial time complexity.**
2. Compute $\mathrm{Atop}(L)$ using backtracking methods - mostly non-polynomial time complexity.

System of partitions

$$\mathfrak{P} = \left(\mathcal{P}_{\text{row}}, \mathcal{P}_{\text{col}}, \mathcal{P}_{\text{sym}} \right)$$

System of partitions

$$\mathfrak{P} = \left(\mathcal{P}_{\mathrm{row}}, \mathcal{P}_{\mathrm{col}}, \mathcal{P}_{\mathrm{sym}} \right)$$

$\mathfrak{P}$ is invariant if and only if each of its components is invariant under $\mathrm{Atop}(L)$

System of partitions

$$\mathfrak{P} = \left(\mathcal{P}_{\text{row}}, \mathcal{P}_{\text{col}}, \mathcal{P}_{\text{sym}}\right)$$

$\mathfrak{P}$ is invariant if and only if each of its components is invariant under $\mathrm{Atop}(L)$

Example:

the types system $\mathfrak{P}_{\mathrm{T}} = \left(\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3\right)$:

$\mathcal{P}_1$ is defined by the number of entries in the rows

$\mathcal{P}_2$ is defined by the number of entries in the columns

$\mathcal{P}_3$ is defined by the number of appearances of each symbol

How to obtain invariant systems?

How to obtain invariant systems?

Aim: find the finest possible invariant
system of partitions

How to obtain invariant systems?

Aim: find the finest possible invariant system of partitions

How?

1) Start with $\mathfrak{P} = \mathfrak{P}_S(L)$ the trivial system of partitions
2) Apply refinement methods

The natural refinement

$\mathfrak{P} = \left(\mathcal{P}_{\mathrm{row}}, \mathcal{P}_{\mathrm{col}}, \mathcal{P}_{\mathrm{sym}}\right)$ induces a partition $E\left(\mathfrak{P}\right)$ on $\mathrm{Ent}(L)$:

The natural refinement

$\mathfrak{P} = \left(\mathcal{P}_{\text{row}}, \mathcal{P}_{\text{col}}, \mathcal{P}_{\text{sym}}\right)$ induces a partition $E\left(\mathfrak{P}\right)$ on $\mathrm{Ent}(L)$:

$$(i, j, L[i, j])) \sim_{E(\mathfrak{P})} (i', j', L[i', j']) \Leftrightarrow \begin{cases} i \sim_{\mathcal{P}_{\text{row}}} i' \\ j \sim_{\mathcal{P}_{\text{col}}} j' \\ L[i, j] \sim_{\mathcal{P}_{\text{sym}}} L[i', j'] \end{cases}$$

The natural refinement

$\mathfrak{P} = \left(\mathcal{P}_{\mathrm{row}}, \mathcal{P}_{\mathrm{col}}, \mathcal{P}_{\mathrm{sym}}\right)$ induces a partition $E\left(\mathfrak{P}\right)$ on $\mathrm{Ent}(L)$:

$$(i, j, L[i, j])) \sim_{E(\mathfrak{P})} (i', j', L[i', j']) \Leftrightarrow \begin{cases} i \sim_{\mathcal{P}_{\mathrm{row}}} i' \\ j \sim_{\mathcal{P}_{\mathrm{col}}} j' \\ L[i, j] \sim_{\mathcal{P}_{\mathrm{sym}}} L[i', j'] \end{cases}$$

1) Label the elements of $\mathrm{Ent}(L)$ by their part in $E\left(\mathfrak{P}\right)$

The natural refinement

$\mathfrak{P} = \left(\mathcal{P}_{\text{row}}, \mathcal{P}_{\text{col}}, \mathcal{P}_{\text{sym}}\right)$ induces a partition $E(\mathfrak{P})$ on $\mathrm{Ent}(L)$:

$$(i, j, L[i, j])) \sim_{E(\mathfrak{P})} (i', j', L[i', j']) \Leftrightarrow \begin{cases} i \sim_{\mathcal{P}_{\text{row}}} i' \\ j \sim_{\mathcal{P}_{\text{col}}} j' \\ L[i, j] \sim_{\mathcal{P}_{\text{sym}}} L[i', j'] \end{cases}$$

1) Label the elements of $\mathrm{Ent}(L)$ by their part in $E(\mathfrak{P})$

2) Define the natural refinement $N(\mathfrak{P}) = \left(N(\mathcal{P}_{\text{row}}), N(\mathcal{P}_{\text{col}}), N(\mathcal{P}_{\text{sym}})\right)$ where

$N(\mathcal{P}_{\text{row}})$ is define by the multisets of labels in the rows

$N(\mathcal{P}_{\text{col}})$ is define by the multisets of labels in the columns

$N(\mathcal{P}_{\text{col}})$ is define by the multisets of labels corresponding to the symbols

The natural refinement

The natural refinement

- $N(\mathfrak{P}) \leq \mathfrak{P}$
-

The natural refinement

- $N(\mathfrak{P}) \leq \mathfrak{P}$
- If $\mathfrak{P}$ is invariant so is $N(\mathfrak{P})$

The natural refinement

- $N(\mathfrak{P}) \le \mathfrak{P}$
- If $\mathfrak{P}$ is invariant so is $N(\mathfrak{P})$
- If $\mathfrak{P}' \le \mathfrak{P}$ then $N(\mathfrak{P}') \le N(\mathfrak{P})$

The natural refinement

- $N(\mathfrak{P}) \leq \mathfrak{P}$
- If $\mathfrak{P}$ is invariant so is $N(\mathfrak{P})$
- If $\mathfrak{P}' \leq \mathfrak{P}$ then $N(\mathfrak{P}') \leq N(\mathfrak{P})$

Can continue the process until it stops:

$$N[\mathfrak{P}] \leq \ldots \leq N\left(N(\mathfrak{P})\right) \leq N(\mathfrak{P}) \leq \mathfrak{P},$$

The natural refinement

- $N(\mathfrak{P}) \leq \mathfrak{P}$
- If $\mathfrak{P}$ is invariant so is $N(\mathfrak{P})$
- If $\mathfrak{P}' \leq \mathfrak{P}$ then $N(\mathfrak{P}') \leq N(\mathfrak{P})$

Can continue the process until it stops:

$$N[\mathfrak{P}] \leq \ldots \leq N\left(N(\mathfrak{P})\right) \leq N(\mathfrak{P}) \leq \mathfrak{P},$$

Examples: $N(\mathfrak{P}_\mathrm{S}) = \mathfrak{P}_\mathrm{T}$ $\quad N(N(\mathfrak{P}_\mathrm{S})) = \mathfrak{P}_\mathrm{SEI}$

the types partition

the Strong Entry Invariants partition (Falcon and Stones, 2017)

The natural refinement

- $N(\mathfrak{P}) \leq \mathfrak{P}$
- If $\mathfrak{P}$ is invariant so is $N(\mathfrak{P})$
- If $\mathfrak{P}' \leq \mathfrak{P}$ then $N(\mathfrak{P}') \leq N(\mathfrak{P})$

Can continue the process until it stops:

$$N[\mathfrak{P}] \leq \ldots \leq N(N(\mathfrak{P})) \leq N(\mathfrak{P}) \leq \mathfrak{P},$$

Examples: $N(\mathfrak{P}_{\mathrm{S}}) = \mathfrak{P}_{\mathrm{T}}$ $N(N(\mathfrak{P}_{\mathrm{S}})) = \mathfrak{P}_{\mathrm{SEI}}$

the types partition

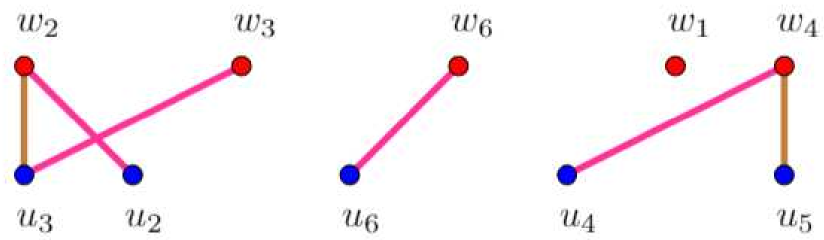
the Strong Entry Invariants partition (Falcon and Stones, 2017)

Disadvantage: when starting from $\mathfrak{P}_{\mathrm{S}}$ it is useless for very dense PLRs

Two-line graphs

for two rows r_1, r_2 in a PLR, construct a vertex-and-edge-colored bipartite graph $G_{r_1,r_2}(L)$ as illustrated here:

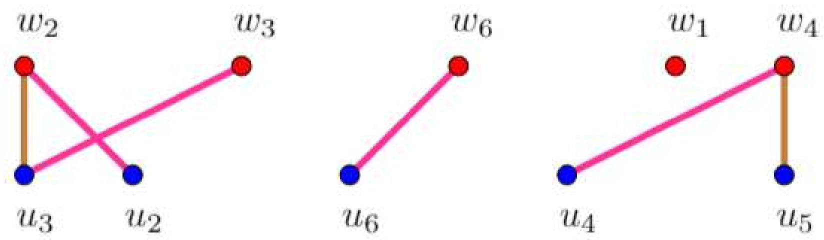
$$L = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 2 & \cdot & 3 & \cdot & 6 & \cdot & 1 & 4 \\ \hline 3 & 2 & \cdot & 6 & \cdot & 4 & \cdot & 5 \\ \hline \end{array} \Rightarrow$$



Two-line graphs

for two rows r_1, r_2 in a PLR, construct a vertex-and-edge-colored bipartite graph $G_{r_1,r_2}(L)$ as illustrated here:

$$L = \begin{array}{|c|c|c|c|c|c|c|c|}\hline 2 & \cdot & 3 & \cdot & 6 & \cdot & 1 & 4 \\ \hline 3 & 2 & \cdot & 6 & \cdot & 4 & \cdot & 5 \\ \hline \end{array}$$



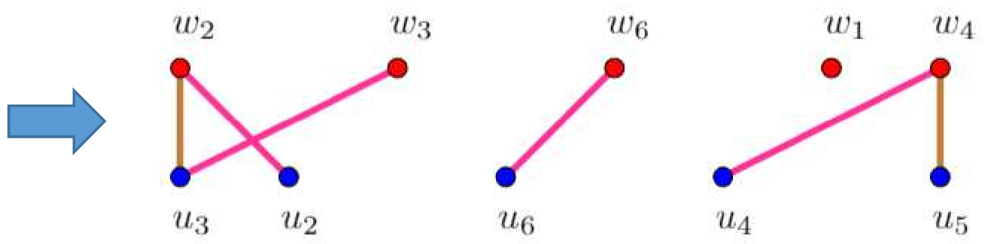
Similar constructions for two columns and two symbols.

(for symbols consider $L^{(1,3)}$, the PLR obtained by swapping the 1st and 3rd coordinates in $\mathrm{Ent}(L)$)

Two-line graphs

for two rows r_1, r_2 in a PLR, construct a vertex-and-edge-colored bipartite graph $G_{r_1,r_2}(L)$ as illustrated here:

$$L = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 2 & \cdot & 3 & \cdot & 6 & \cdot & 1 & 4 \\ \hline 3 & 2 & \cdot & 6 & \cdot & 4 & \cdot & 5 \\ \hline \end{array}$$



Similar constructions for two columns and two symbols.

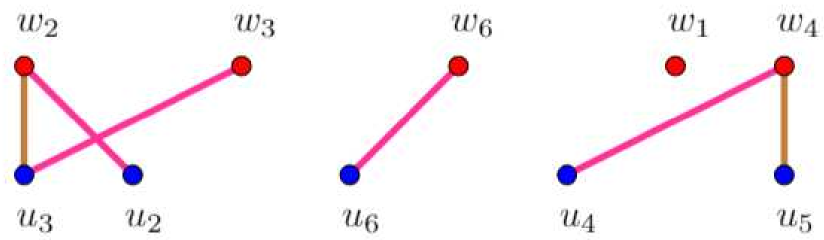
(for symbols consider $L^{(1,3)}$, the PLR obtained by swapping the 1st and 3rd coordinates in $\mathrm{Ent}(L)$)

Can try it here: http://plr.telhai.ac.il/Home/VisualSimulation

Two-line graphs

for two rows r_1, r_2 in a PLR, construct a vertex-and-edge-colored bipartite graph $G_{r_1,r_2}(L)$ as illustrated here:

$$L = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 2 & \cdot & 3 & \cdot & 6 & \cdot & 1 & 4 \\ \hline 3 & 2 & \cdot & 6 & \cdot & 4 & \cdot & 5 \\ \hline \end{array}$$



Similar constructions for two columns and two symbols.

(for symbols consider $L^{(1,3)}$, the PLR obtained by swapping the 1st and 3rd coordinates in $\mathrm{Ent}(L)$)

Can try it here: http://plr.telhai.ac.il/Home/VisualSimulation

Remark: can be viewed as a generalization of cycles in partial permutations

The two-line representations

$$\mathcal{R}_{\text{row}}(L) = \begin{bmatrix} 0 & & & & \\ & 0 & & & \\ & & \ddots & r_{kl} & \\ & & r_{ij} & & \\ & & & & 0 \end{bmatrix}$$

The two-line representations

$$r_{ij} = r_{kl} \quad \Leftrightarrow \quad G_{i,j} \simeq G_{k,l}$$

$$\mathcal{R}_{\text{row}}(L) = \begin{bmatrix} 0 & & & & \\ & 0 & & & \\ & & \ddots & r_{kl} & \\ & & r_{ij} & & \\ & & & & 0 \end{bmatrix}$$

The two-line representations

$$r_{ij} = r_{kl} \quad \Leftrightarrow \quad G_{i,j} \simeq G_{k,l}$$

$$\mathcal{R}_{\mathrm{row}}(L) = \begin{bmatrix} 0 & & & & \\ & 0 & & & \\ & & \ddots & r_{kl} & \\ & & r_{ij} & & \\ & & & & 0 \end{bmatrix}$$

(Define $\mathcal{R}_{\mathrm{col}}(L)$ and $\mathcal{R}_{\mathrm{sym}}(L)$ analogously)

The two-line graph (TLG) refinement

The two-line graph (TLG) refinement

$\mathfrak{P} = \left(\mathcal{P}_{\mathrm{r}}, \mathcal{P}_{\mathrm{c}}, \mathcal{P}_{s} \right)$ an adequate system

The two-line graph (TLG) refinement

$\mathfrak{P} = \left(\mathcal{P}_{\mathrm{r}}, \mathcal{P}_{\mathrm{c}}, \mathcal{P}_{s}\right)$ an adequate system

suppose $\mathcal{P}_{\mathrm{r}} = \{P_1, P_2, \ldots, P_k\}$

The two-line graph (TLG) refinement

$\mathfrak{P} = \left(\mathcal{P}_{\mathrm{r}}, \mathcal{P}_{\mathrm{c}}, \mathcal{P}_{s}\right)$ an adequate system

suppose $\mathcal{P}_{\mathrm{r}} = \{P_1, P_2, \ldots, P_k\}$

$$\mathcal{R}_{\mathrm{row}}(L) = \left[\quad \right]$$

The two-line graph (TLG) refinement

$\mathfrak{P} = \left(\mathcal{P}_{\mathrm{r}}, \mathcal{P}_{\mathrm{c}}, \mathcal{P}_{s}\right)$ an adequate system

suppose $\mathcal{P}_{\mathrm{r}} = \{P_1, P_2, \ldots, P_k\}$

$$\mathcal{R}_{\mathrm{row}}(L) = \begin{array}{c} \begin{array}{cccc} P_1 & P_2 & \quad P_i \quad & P_k \end{array} \\ \left[\begin{array}{c|c|c|c} \quad & \quad & \quad \cdots \quad & \quad \end{array}\right] \end{array}$$

The two-line graph (TLG) refinement

$\mathfrak{P} = (\mathcal{P}_{\mathrm{r}}, \mathcal{P}_{\mathrm{c}}, \mathcal{P}_{s})$ an adequate system

suppose $\mathcal{P}_{\mathrm{r}} = \{P_1, P_2, \ldots, P_k\}$

$$\begin{array}{cc} & \begin{array}{cccc} P_1 & P_2 & P_i & P_k \end{array} \\ \mathcal{R}_{\mathrm{row}}(L) = \begin{array}{c} r_1 \\ r_2 \end{array} & \left[\begin{array}{c|c|c|c} & & \cdots & \\ & & & \end{array}\right] \end{array}$$

$$r_1 \sim_{G(\mathcal{P}_{\mathrm{r}})} r_2 \quad \Leftrightarrow \quad \begin{cases} r_1 \sim_{\mathcal{P}_{\mathrm{r}}} r_2 \\ \text{the multisets agree in each part} \end{cases}$$

The two-line graph (TLG) refinement

$\mathfrak{P} = (\mathcal{P}_{\mathrm{r}}, \mathcal{P}_{\mathrm{c}}, \mathcal{P}_{s})$ an adequate system

suppose $\mathcal{P}_{\mathrm{r}} = \{P_1, P_2, \ldots, P_k\}$

$$\mathcal{R}_{\mathrm{row}}(L) = \begin{array}{c|c|c|c|c|c} & P_1 & P_2 & P_i & & P_k \\ r_1 & 1\ 1\ 3\ 2 & 4\ 2\ 2 & \cdots & & 6\ \ 12\ \ 9 \\ r_2 & 3\ 1\ 1\ 2 & 2\ 4\ 2 & & & 12\ \ 9\ \ 6 \end{array}$$

$r_1 \sim_{G(\mathcal{P}_{\mathrm{r}})} r_2 \iff \begin{cases} r_1 \sim_{\mathcal{P}_{\mathrm{r}}} r_2 \\ \text{the multisets agree in each part} \end{cases}$

The two-line graph (TLG) refinement

$\mathfrak{P} = (\mathcal{P}_{\mathrm{r}}, \mathcal{P}_{\mathrm{c}}, \mathcal{P}_{s})$ an adequate system

suppose $\mathcal{P}_{\mathrm{r}} = \{P_1, P_2, \ldots, P_k\}$

$$\mathcal{R}_{\mathrm{row}}(L) = \begin{array}{c} \\ r_1 \\ r_2 \end{array} \begin{array}{c|c|c|c} P_1 & P_2 & P_i & P_k \\ \hline 1\ 1\ 3\ 2 & 4\ 2\ 2 & \cdots & 6\ \ 12\ \ 9 \\ 3\ 1\ 1\ 2 & 2\ 4\ 2 & & 12\ \ 9\ \ 6 \end{array}$$

$$r_1 \sim_{G(\mathcal{P}_{\mathrm{r}})} r_2 \quad \Leftrightarrow \quad \begin{cases} r_1 \sim_{\mathcal{P}_{\mathrm{r}}} r_2 \\ \text{the multisets} \\ \text{agree in each part} \end{cases}$$

analogous definitions for $G(\mathcal{P}_{\mathrm{c}})$ and $G(\mathcal{P}_{\mathrm{s}})$

$$G(\mathfrak{P}) = \big(G(\mathcal{P}_{\mathrm{r}}), G(\mathcal{P}_{\mathrm{c}}), G(\mathcal{P}_{s})\big)$$

The TLG refinement

The TLG refinement

- $G(\mathfrak{P}) \leq \mathfrak{P}$
-

The TLG refinement

- $G(\mathfrak{P}) \leq \mathfrak{P}$
- If $\mathfrak{P}$ is invariant so is $G(\mathfrak{P})$

The TLG refinement

- $G(\mathfrak{P}) \leq \mathfrak{P}$
- If $\mathfrak{P}$ is invariant so is $G(\mathfrak{P})$
- If $\mathfrak{P}' \leq \mathfrak{P}$ then $G(\mathfrak{P}') \leq G(\mathfrak{P})$

The TLG refinement

- $G(\mathfrak{P}) \leq \mathfrak{P}$
- If $\mathfrak{P}$ is invariant so is $G(\mathfrak{P})$
- If $\mathfrak{P}' \leq \mathfrak{P}$ then $G(\mathfrak{P}') \leq G(\mathfrak{P})$

Can continue the process until it stops:

$$G[\mathfrak{P}] \leq \ldots \leq G\big(G(\mathfrak{P})\big) \leq G(\mathfrak{P}) \leq \mathfrak{P},$$

G vs. N

Not much is known about the relation between the refinements G and N. However, it is easy to see

G vs. N

Not much is known about the relation between the refinements G and N. However, it is easy to see

$$G(\mathfrak{P}_{\mathrm{S}}) \leq N(\mathfrak{P}_{\mathrm{S}})$$

G vs. N

Not much is known about the relation between the refinements G and N. However, it is easy to see

$$G(\mathfrak{P}_S) \le N(\mathfrak{P}_S)$$

$$\Downarrow$$

$$G[\mathfrak{P}_S] \le N[\mathfrak{P}_S]$$

G vs. N

Not much is known about the relation between the refinements G and N. However, it is easy to see

$$G(\mathfrak{P}_S) \leq N(\mathfrak{P}_S)$$

$$\Downarrow$$

$$G\left[\mathfrak{P}_S\right] \leq N\left[\mathfrak{P}_S\right]$$

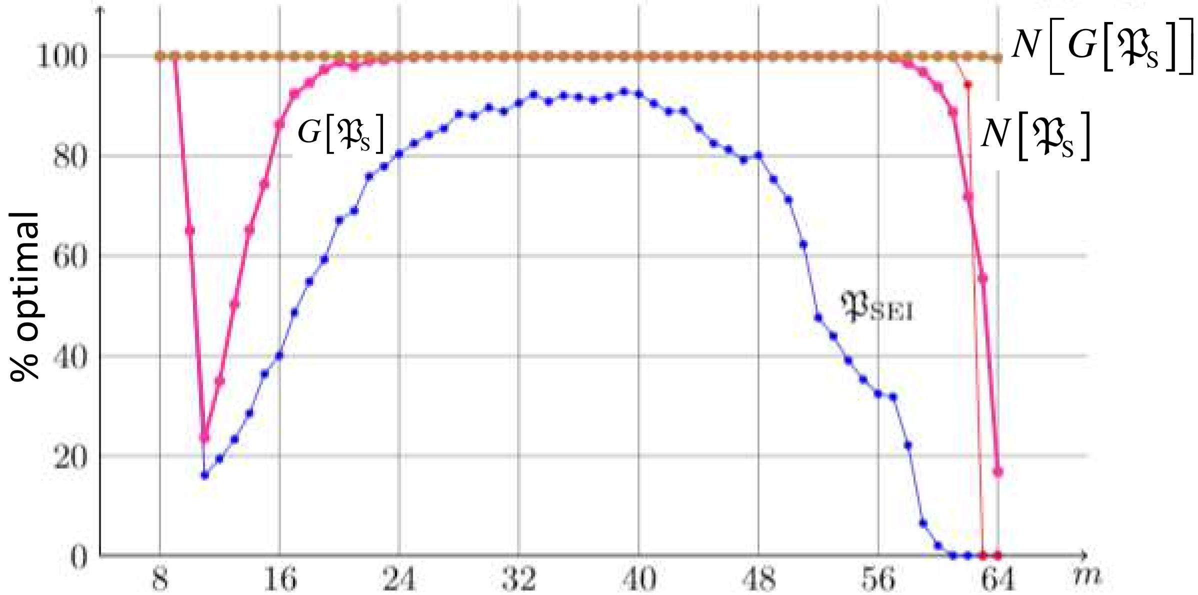
This does not mean that the natural refinement is redundant. It is possible to have

$$N\left(G[\mathfrak{P}]\right) < G[\mathfrak{P}]$$

Complexity

The average complexity of the TLG refinement for $L \in \mathrm{PLR}(r, s, n)$ is $\mathcal{O}(M^3 \log M)$, where $M = \max(r, s, n)$

Performance on random PLRs



1000 random PLRs in PLR(8,8,8;m) for each m

Works well for dense PLRs and (full) Latin rectangles.

Problems for further study

Problems for further study

- What is the best combination of the operators N and G (in terms of best refinement and lowest complexity)?

Problems for further study

- What is the best combination of the operators N and G (in terms of best refinement and lowest complexity)?
- Find other refinement methods.

Problems for further study

- What is the best combination of the operators N and G (in terms of best refinement and lowest complexity)?
- Find other refinement methods.
- What is the most efficient way to conduct the subsequent search?

T	•	H	•	A	•	N	K	•
•	•	•	Y	O	•	U	•	
•	F	•	O	•	R	•	•	•
•	Y	O	•	•	U	•	R	•
•	T	•	•	•	•	•	•	•
A	•	T	E	N	•	I	O	N
•	•	•	•	•	T	•	•	•