An upper bound of 84 for Morpion Solitaire 5D

Henryk Michalewski, Andrzej Nagórko, Jakub Pawlewicz

University of Warsaw

28th Canadian Conference on Computational Geometry

CCCG 2016

Outline

Outline

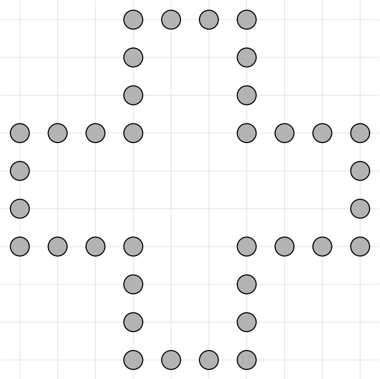
Morpion Solitaire

- A single-player paper-and-pencil game
  The goal is to find longest possible sequence of valid moves.
- Popularized in France in 70's by *Science & Vie* magazine
  The magazine published long sequences found by its readers.
- Two interesting variants: 5T and 5D
- Difficult for computers

5T: The human record was obtained by C.-H. Bruneau in 1976.
It was beat by NRPA (Monte Carlo) algorithm by Ch. Rosin in 2010.
Rosin got best paper award at IJCAI 2011 for this work.

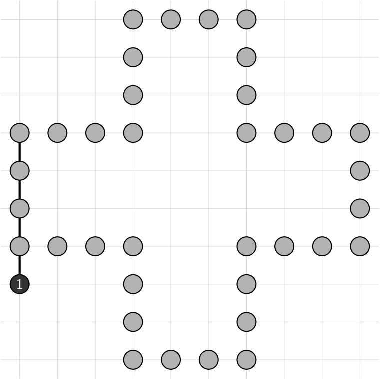
5D: The human record was obtained by A. Langerman in 1999.
Beaten by H. Hyyrö and T. Poranen in 2006.
Rosin set the world record in 2011.

Rules of Morpion Solitaire



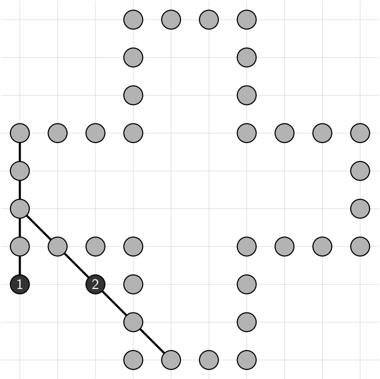
Initial position: 36 dots arranged in a cross on a square grid.

Rules of Morpion Solitaire

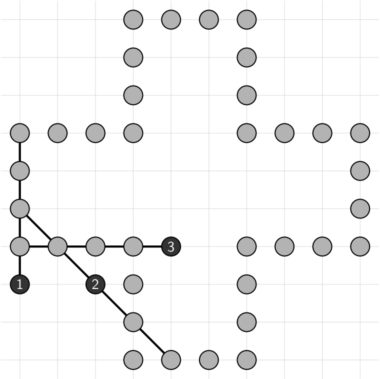


A move: place a dot and draw a line (diagonal, horizontal or vertical) passing through the placed dot and four others.

Rules of Morpion Solitaire

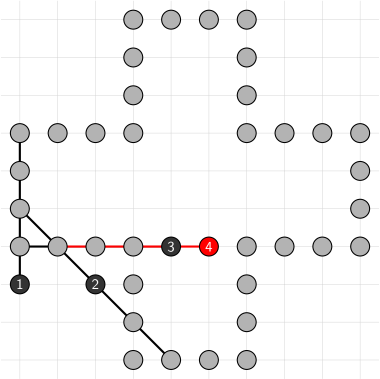


Two lines may cross.



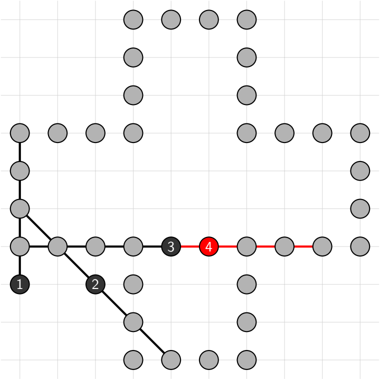
Vertical, diagonal and horizontal moves allowed.

Rules of Morpion Solitaire



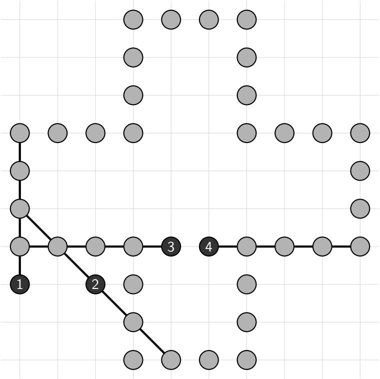
Two parallel moves may not overlap,

Rules of Morpion Solitaire



neither touch (however allowed in 5T variant),

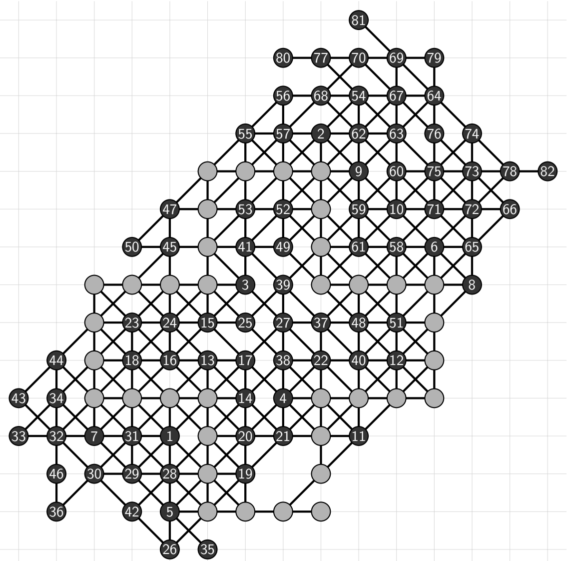
Rules of Morpion Solitaire



but allowed if disjoint.

The goal: find the longest sequence of moves

82 moves
Rosin 2011



Upper bound

The potential method gives a bound of 144 moves by Demaine et al. (2006). A geometric analysis improves the upper bound to 121 moves by Kawamura et al. (2013).

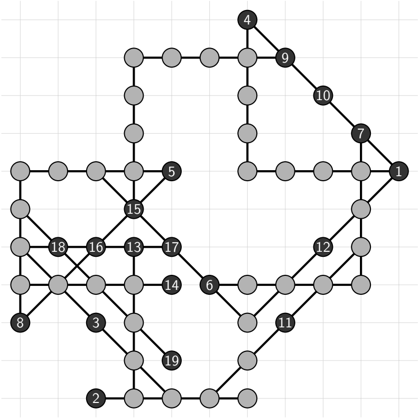
We prove an upper bound of 84.

The proof has two main ingredients:

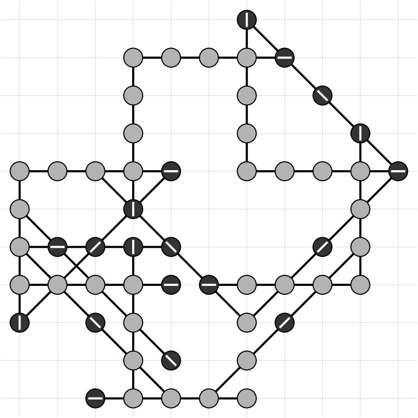
- modelling the game by linear programming,
- limiting the size of the board by resizing process.

Outline

Morpion 5D position and marked graph



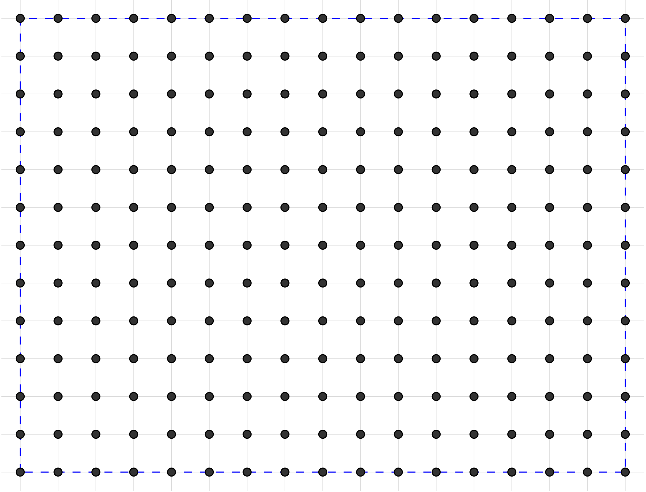
Morpion 5D position



Marked Morpion 5D graph

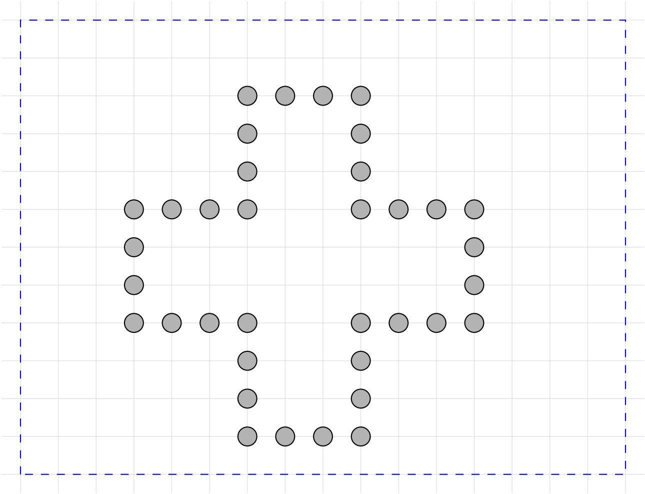
Every Morpion 5D position has corresponding marked Morpion 5D graph.

The board



We limit the board to rectangular grid $\mathcal{B} \subset \mathbb{Z}^2$ of points where the game is played.

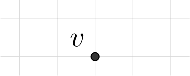
The board



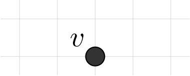
$\texttt{Cross} \subset \mathcal{B}$ denote dots from the initial cross.

Variables

1. dot_v, $v \in \mathcal{B}$.



$\mathrm{dot}_v = 0$



$\mathrm{dot}_v = 1$

Desired property: $\mathrm{dot}_v = 1$ iff dot was put at point v.

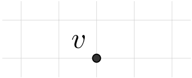
2. $\mathrm{move}_{m,v}$, m is a move, v is a played dot (or marking of m).
Move m is represented by five consecutive points on a horizontal, vertical or diagonal line: $m = \{d_1, d_2, d_3, d_4, d_5\}$, $v \in m$.

$\mathrm{move}_{m,v} = 0$ $\qquad$ $\mathrm{move}_{m,v} = 0$ $\qquad$ $\mathrm{move}_{m,v} = \begin{cases} 1 & \text{if } v = d_4 \\ 0 & \text{if } v \neq d_4 \end{cases}$

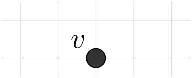
Desired property: $\mathrm{move}_{m,v} = 1$ iff a move was played by putting dot at v and drawing line through points from set m.

Variables

1. dot_v, $v \in \mathcal{B}$.



$\mathrm{dot}_v = 0$



$\mathrm{dot}_v = 1$

Desired property: $\mathrm{dot}_v = 1$ iff dot was put at point v.

2. $\mathrm{move}_{m,v}$, m is a move, v is a played dot (or marking of m). *Move* m is represented by five consecutive points on a horizontal, vertical or diagonal line: $m = \{d_1, d_2, d_3, d_4, d_5\}$, $v \in m$.

$\mathrm{move}_{m,v} = 0$

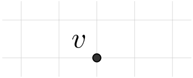
$\mathrm{move}_{m,v} = 0$

$$\mathrm{move}_{m,v} = \begin{cases} 1 & \text{if } v = d_4 \\ 0 & \text{if } v \neq d_4 \end{cases}$$

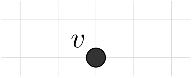
Desired property: $\mathrm{move}_{m,v} = 1$ iff a move was played by putting dot at v and drawing line through points from set m.

Variables

1. dot_v, $v \in \mathcal{B}$.



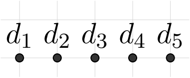
$\mathrm{dot}_v = 0$



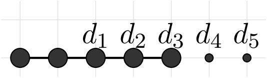
$\mathrm{dot}_v = 1$

Desired property: $\mathrm{dot}_v = 1$ iff dot was put at point v.

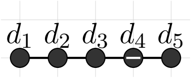
2. $\mathrm{move}_{m,v}$, m is a move, v is a played dot (or marking of m).
Move m is represented by five consecutive points on a horizontal, vertical or diagonal line: $m = \{d_1, d_2, d_3, d_4, d_5\}$, $v \in m$.



$\mathrm{move}_{m,v} = 0$



$\mathrm{move}_{m,v} = 0$



$$\mathrm{move}_{m,v} = \begin{cases} 1 & \text{if } v = d_4 \\ 0 & \text{if } v \neq d_4 \end{cases}$$

Desired property: $\mathrm{move}_{m,v} = 1$ iff a move was played by putting dot at v and drawing line through points from set m.

Constraints: initial cross

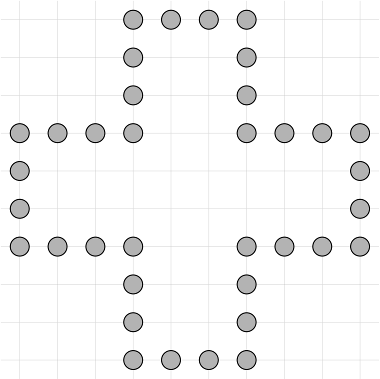
Let $v \in \texttt{Cross}$

Dots of cross must be put:

L_1	$\mathrm{dot}_v = 1$

v cannot be put by a move:

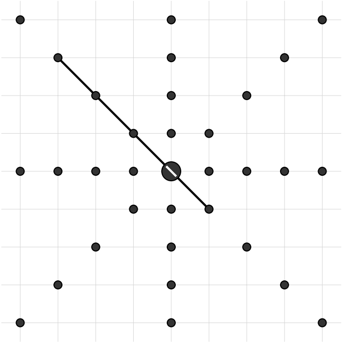
L_2	$\mathrm{move}_{m,v} = 0$ for all m s.t. $v \in m$



Constraints: put dot

Let $v \in \mathcal{B} \setminus \texttt{Cross}$.
If dot v is put, there must be exactly one move by which it was put:

L3
$$\mathrm{dot}_v = \sum_{m \text{ s.t. } v \in m} \mathrm{move}_{m,v}$$

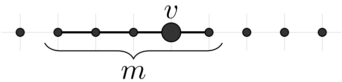


Constraints: conflicting moves, move needs dots

Let $v \in \mathcal{B}$ let d be a direction (vertical, horizontal or one of diagonals).
No move in the same direction should share a dot v:

$$\mathrm{L}_4 \qquad \mathrm{dot}_v \geq \sum_{\substack{m \text{ s.t. } v \in m \\ m \text{ has direction } d}} \mathrm{move}_m \quad \text{where } \mathrm{move}_m = \sum_{w \in m} \mathrm{move}_{m,w}$$

move_m says whether a move through dots m was played.



L_4 also enforces a condition:

$$\mathrm{move}_m \leq \mathrm{dot}_v \quad \text{for each } v \in m$$

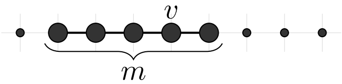
i.e. if move m is played each dot $v \in m$ must be put.

Constraints: conflicting moves, move needs dots

Let $v \in \mathcal{B}$ let d be a direction (vertical, horizontal or one of diagonals). No move in the same direction should share a dot v:

$$\mathsf{L}_4 \qquad \mathrm{dot}_v \geq \sum_{\substack{m \text{ s.t. } v \in m \\ m \text{ has direction } d}} \mathrm{move}_m \quad \text{where } \mathrm{move}_m = \sum_{w \in m} \mathrm{move}_{m,w}$$

move_m says whether a move through dots m was played.

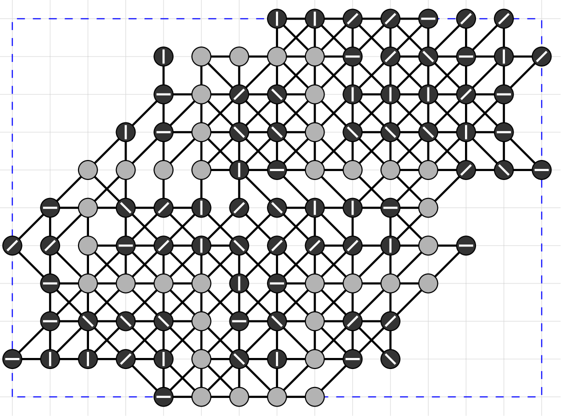


L_4 also enforces a condition:

$$\mathrm{move}_m \leq \mathrm{dot}_v \quad \text{for each } v \in m$$

i.e. if move m is played each dot $v \in m$ must be put.

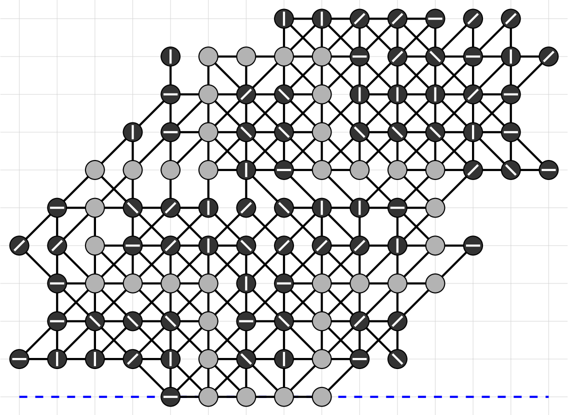
Constraints: fitting the board



For resizing process a solution should fit $\mathcal{B}$. For each side $S \subset \mathcal{B}$ we require

L5 $$\sum_{v \in S} \mathrm{dot}_v \geq 1$$

Constraints: fitting the board



For resizing process a solution should fit $\mathcal{B}$. For each side $S \subset \mathcal{B}$ we require

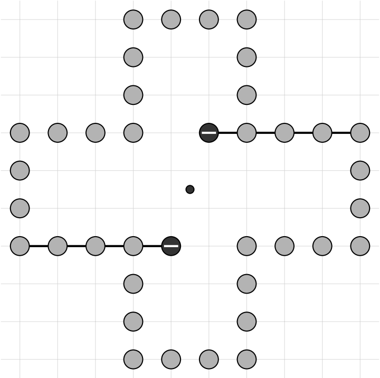
L5
$$\sum_{v \in S} \mathrm{dot}_v \geq 1$$

Additional constraints: symmetric games

Let m be a move and $v \in m$. Let m_s, v_s are symmetric to m, v with respect of the centre of the `Cross`. For symmetric games we have:

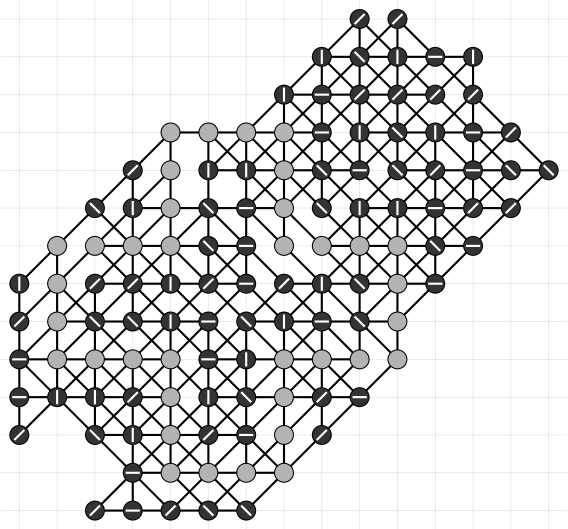
L_6

$$\mathrm{move}_{m,v} = \mathrm{move}_{m_s,v_s}$$



Additional constraints: move ordering

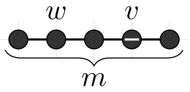
Marked Morpion 5D graph $\not\Rightarrow$ Morpion 5D position



Additional constraints: move ordering

Additional variable:

3. ord_v, $v \in \mathcal{B}$, continuous.
   Desired property: $\mathrm{ord}_v \geq \mathrm{ord}_w + 1$ if a move m put dot v and $w \in m \setminus \{v\}$.



Constraint enforcing order on moves:

L7 $\quad \mathrm{ord}_v \geq \mathrm{ord}_w + 1 - 121(1 - \mathrm{move}_{m,v})$ for each $w \in m \setminus \{v\}$

Objective

Maximize the number of moves:

Obj maximize $\sum_{m} \sum_{v \in m} \mathrm{move}_{m,v}$

Outline

Feasible and infeasible boxes

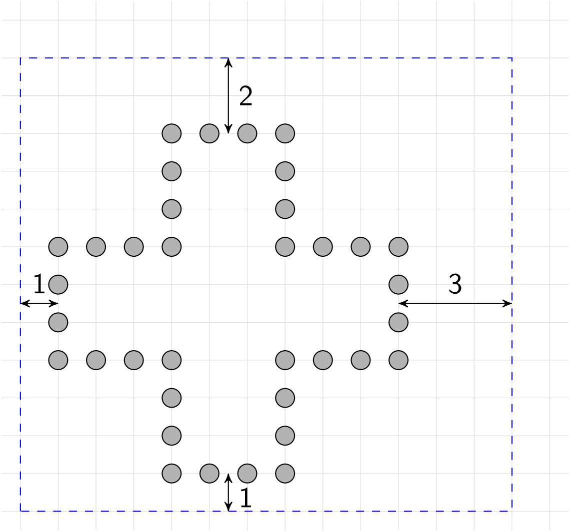
Feasible box

Box $\mathcal{B}$ is *feasible* if mixed integer linear program with conditions L_1–L_5 is feasible, i.e. there exists marked Morpion 5D graph fitting box $\mathcal{B}$.

Infeasible box

Box $\mathcal{B}$ is *infeasible* if it is not feasible.

Resized box



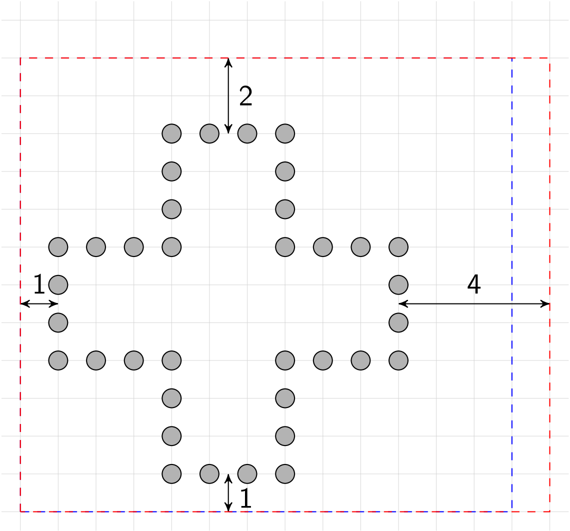
Box by distances

We describe a box by a 4-tuple of distances from `Cross`: $(3, 2, 1, 1)$.

Resized box

Box $\mathcal{B}$ is *resized* from box $\mathcal{B}'$ if one side is extended by 1 or two neighboring sides are extended by 1.

Resized box



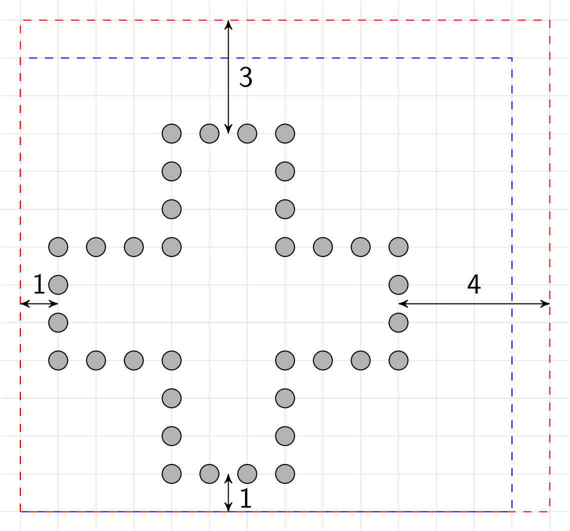
Box by distances

We describe a box by a 4-tuple of distances from `Cross`: $(3, 2, 1, 1)$.

Resized box

Box $\mathcal{B}$ is *resized* from box $\mathcal{B}'$ if one side is extended by 1 or two neighboring sides are extended by 1.

Resized box



Box by distances

We describe a box by a 4-tuple of distances from `Cross`: $(3, 2, 1, 1)$.

Resized box

Box $\color{red}{\mathcal{B}}$ is *resized* from box $\color{blue}{\mathcal{B}'}$ if one side is extended by 1 or two neighboring sides are extended by 1.

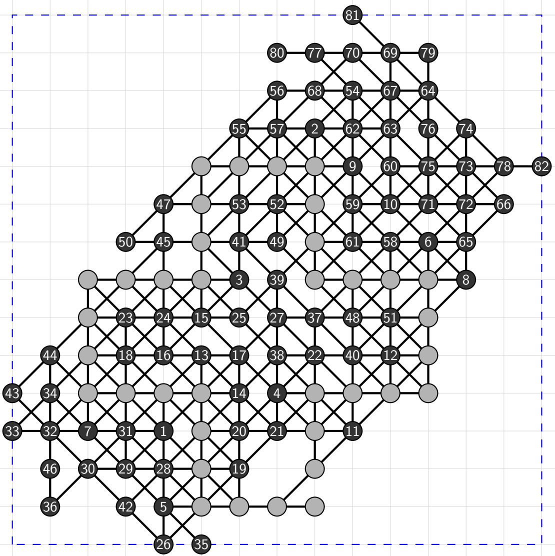
Resizing process

Observation

Let $\mathcal{B}$ be a box of a Morpion 5D position. There exists $\mathcal{B}'$ s.t. $\mathcal{B}$ is resized from $\mathcal{B}'$ and there exists Morpion 5D position fitting box $\mathcal{B}'$.

Corollary

If $\mathcal{B}$ is a box of a Morpion 5D position, there exists a sequence of feasible boxes $(0,0,0,0) = \mathcal{B}_0, \ldots, \mathcal{B}_n = \mathcal{B}$, s.t. $\mathcal{B}_{k+1}$ is resized from $\mathcal{B}_k$.



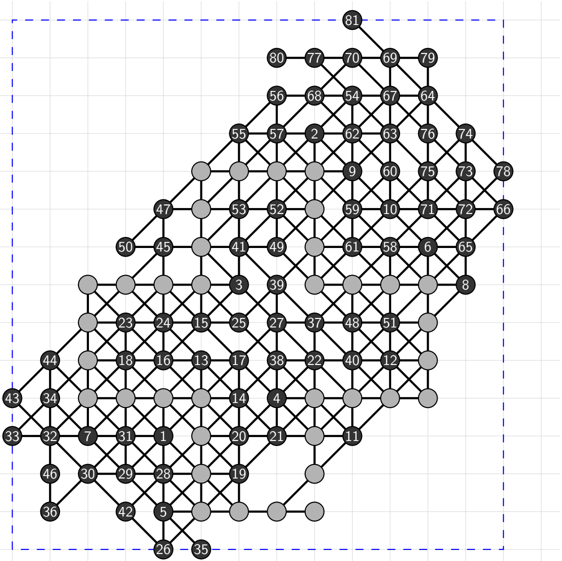
Resizing process

Observation

Let $\mathcal{B}$ be a box of a Morpion 5D position. There exists $\mathcal{B}'$ s.t. $\mathcal{B}$ is resized from $\mathcal{B}'$ and there exists Morpion 5D position fitting box $\mathcal{B}'$.

Corollary

If $\mathcal{B}$ is a box of a Morpion 5D position, there exists a sequence of feasible boxes $(0,0,0,0) = \mathcal{B}_0, \ldots, \mathcal{B}_n = \mathcal{B}$, s.t. $\mathcal{B}_{k+1}$ is resized from $\mathcal{B}_k$.

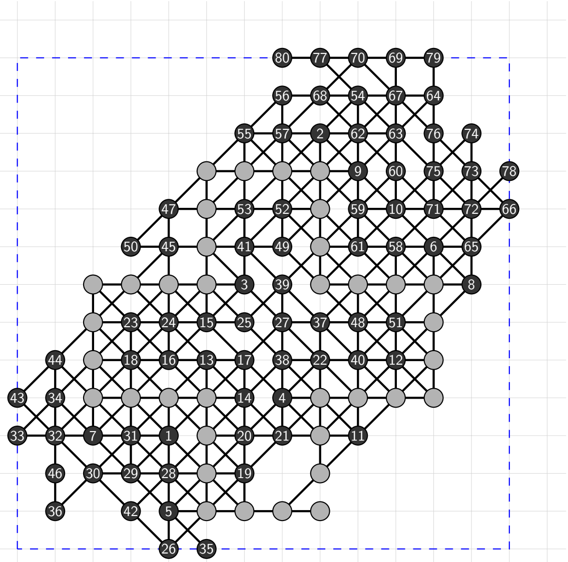


Observation

Let $\mathcal{B}$ be a box of a Morpion 5D position. There exists $\mathcal{B}'$ s.t. $\mathcal{B}$ is resized from $\mathcal{B}'$ and there exists Morpion 5D position fitting box $\mathcal{B}'$.

Corollary

If $\mathcal{B}$ is a box of a Morpion 5D position, there exists a sequence of feasible boxes $(0,0,0,0) = \mathcal{B}_0, \ldots, \mathcal{B}_n = \mathcal{B}$, s.t. $\mathcal{B}_{k+1}$ is resized from $\mathcal{B}_k$.

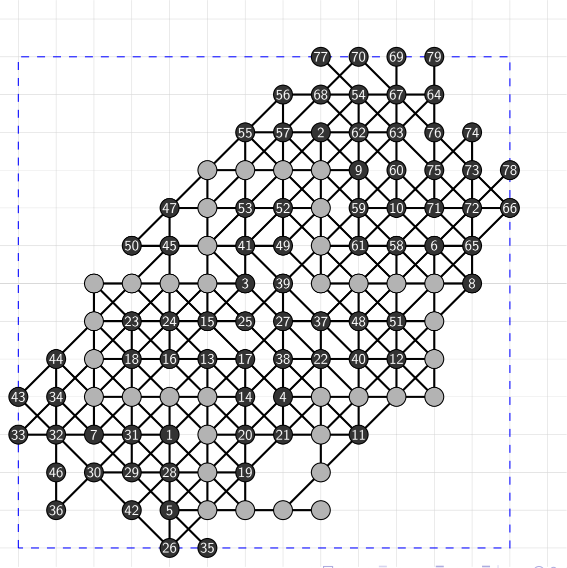


Observation

Let $\mathcal{B}$ be a box of a Morpion 5D position. There exists $\mathcal{B}'$ s.t. $\mathcal{B}$ is resized from $\mathcal{B}'$ and there exists Morpion 5D position fitting box $\mathcal{B}'$.

Corollary

If $\mathcal{B}$ is a box of a Morpion 5D position, there exists a sequence of feasible boxes $(0,0,0,0) = \mathcal{B}_0, \ldots, \mathcal{B}_n = \mathcal{B}$, s.t. $\mathcal{B}_{k+1}$ is resized from $\mathcal{B}_k$.



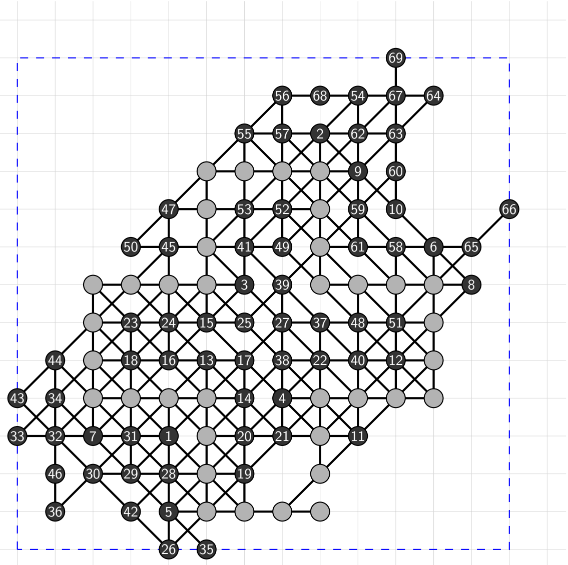
Resizing process

Observation

Let $\mathcal{B}$ be a box of a Morpion 5D position. There exists $\mathcal{B}'$ s.t. $\mathcal{B}$ is resized from $\mathcal{B}'$ and there exists Morpion 5D position fitting box $\mathcal{B}'$.

Corollary

If $\mathcal{B}$ is a box of a Morpion 5D position, there exists a sequence of feasible boxes $(0,0,0,0) = \mathcal{B}_0, \ldots, \mathcal{B}_n = \mathcal{B}$, s.t. $\mathcal{B}_{k+1}$ is resized from $\mathcal{B}_k$.



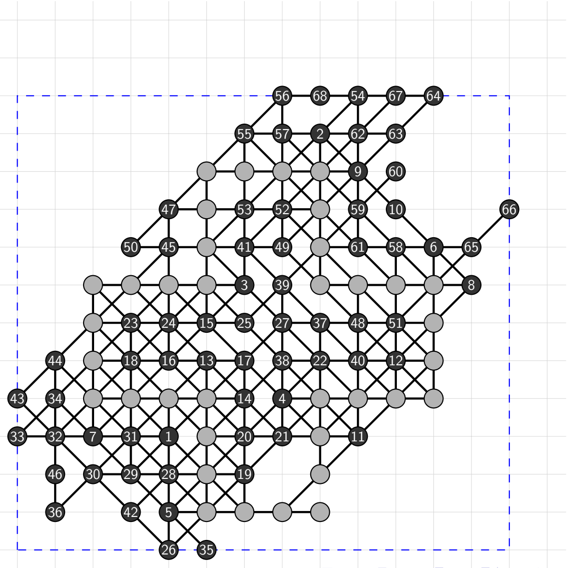
Resizing process

Observation

Let $\mathcal{B}$ be a box of a Morpion 5D position. There exists $\mathcal{B}'$ s.t. $\mathcal{B}$ is resized from $\mathcal{B}'$ and there exists Morpion 5D position fitting box $\mathcal{B}'$.

Corollary

If $\mathcal{B}$ is a box of a Morpion 5D position, there exists a sequence of feasible boxes $(0,0,0,0) = \mathcal{B}_0, \ldots, \mathcal{B}_n = \mathcal{B}$, s.t. $\mathcal{B}_{k+1}$ is resized from $\mathcal{B}_k$.



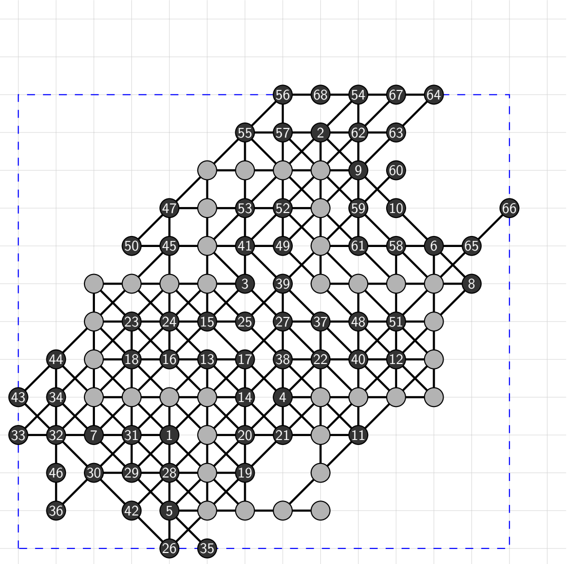
Resizing process

Observation

Let $\mathcal{B}$ be a box of a Morpion 5D position. There exists $\mathcal{B}'$ s.t. $\mathcal{B}$ is resized from $\mathcal{B}'$ and there exists Morpion 5D position fitting box $\mathcal{B}'$.

Corollary

If $\mathcal{B}$ is a box of a Morpion 5D position, there exists a sequence of feasible boxes $(0,0,0,0) = \mathcal{B}_0, \ldots, \mathcal{B}_n = \mathcal{B}$, s.t. $\mathcal{B}_{k+1}$ is resized from $\mathcal{B}_k$.



The algorithm for resizing process

unsolved $\leftarrow \{(0,0,0,0)\}$
solved $\leftarrow \emptyset$
while *unsolved* $\neq \emptyset$ **do**
  $\mathcal{B} \leftarrow$ pop box from *unsolved*
  solved $\leftarrow$ *solved* $\cup \{\mathcal{B}\}$
  if $\textsc{Solve}(\mathcal{B})$ is feasible **then** ▷ Applying linear solver
    for each $\mathcal{B}_{\text{cand}}$ resized from $\mathcal{B}$ **do** ▷ Up to 8 candidates
      if $\mathcal{B}_{\text{cand}} \notin$ *solved* $\cup$ *unsolved* with respect to symmetries **then**
        unsolved $\leftarrow$ *unsolved* $\cup \{\mathcal{B}_{\text{cand}}\}$

Outline

The hardest feasible boxes

No	BBox	Size
1	(4, 3, 1, 1)	85.0
2	(4, 3, 1, 2)	85.0
3	(4, 3, 1, 3)	85.0
4	(4, 2, 1, 2)	84.0
5	(4, 2, 2, 2)	84.0
6	(5, 2, 2, 1)	84.0
7	(5, 2, 1, 2)	84.0
8	(5, 2, 2, 2)	84.0
9	(3, 3, 2, 1)	84.0
10	(3, 3, 2, 2)	84.0
11	(4, 3, 2, 1)	84.0
12	(4, 3, 3, 1)	84.0
13	(4, 3, 2, 2)	84.0
14	(4, 3, 2, 3)	84.0

No	BBox	Size
15	(4, 3, 0, 2)	84.0
16	(3, 2, 1, 2)	83.0
17	(3, 2, 2, 2)	83.0
18	(5, 2, 1, 1)	83.0
19	(3, 3, 3, 1)	83.0
20	(4, 3, 3, 2)	83.0
21	(5, 3, 1, 1)	83.0
22	(5, 3, 1, 2)	83.0
23	(4, 3, 0, 3)	83.0
24	(4, 4, 1, 0)	83.0
25	(4, 4, 2, 0)	83.0
26	(4, 4, 1, 1)	83.0
27	(4, 4, 2, 1)	83.0
28	(4, 4, 3, 1)	83.0

Upper bound of 84

For three bounding boxes with $\mathrm{Size} = 85.0$ we run solver again with added constraint L_7.
It was able to reduce upper bound below 85.

Using the same process we prove that the best score for symmetric Morpion 5D is 68 (found by M. Quist in 2008).

Upper bound of 84

For three bounding boxes with $\mathrm{Size} = 85.0$ we run solver again with added constraint L_7.
It was able to reduce upper bound below 85.

Using the same process we prove that the best score for symmetric Morpion 5D is 68 (found by M. Quist in 2008).

Outline

5 Appendix

Progress in proving upper bound of 82

https://github.com/anagorko/morpion-lpp/wiki/Solving-5D