

State Representation and Polyomino Placement for the Game Patchwork

Mikael Zayenz Lagerkvist

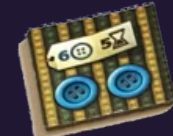
[Game] State
Representation

Polyomino

Placement

the Game

Patchwork



What is Patchwork?

The background of the slide features several components of the board game Patchwork. There are three wooden boards of different stages of completion, scattered wooden tiles with various patterns and numbers, and a pile of small wooden coins. The items are arranged in a somewhat chaotic but organized manner, showcasing the game's components.

- Released 2014 by Uwe Rosenberg
- Popular (place 64 on Board Game Geek)
- 2-player
- Full information
- Experimentally: average 23 plies and branching factor 83

Patchwork rules

The background of the slide features a collection of Patchwork game components. There are three boards of different sizes and designs, some with various patterns and colors. Scattered around the boards are numerous patches of various shapes and sizes, some with numbers and symbols. There are also several buttons of different colors and designs. The entire scene is set against a dark, textured background.

- Central time board
- Buying patches and placing on boards
- Earn buttons and cover the board
- Packing the board well core challenge
 - To buy a patch it must fit on the board

A collection of board game components including boards, pieces, and cards scattered on a dark background. The components are primarily in shades of brown and gold. There are three board pieces, several L-shaped and cross-shaped pieces, and numerous small circular tokens. The text "Game state?" is overlaid in the center in a white, stylized font.

Game state??

Game state



- Representing the current state of a game
- Must support needs of surrounding code
 - Possible moves, evaluation, making moves, ...
- Classical board game engines have hyper-optimized small states
- Modern board games are more complex

The background features a 15-puzzle board and its pieces. The board is a 4x4 grid with a central 2x2 area containing a pattern of squares. The pieces are numbered 1 through 15 and are scattered around the board. The text is overlaid on this background.

Constraint programming to
the rescue!

Use constraint programming for the
placement part

Re-use smart code in solvers

Placement using regular expressions

The background features a 15x15 grid with various pieces and coins scattered around it. The pieces are of different shapes and sizes, some with numbers and symbols on them. The coins are small and round. The grid is partially filled with these pieces, illustrating the placement problem.

- Based on paper with Gilles Pesant from 2008
- Extended with
 - Explicit rotations
 - Reified placement
 - Usage constraints



Propagation finds
all placeable patches

The background features several golden Tetris game boards and various Tetris pieces scattered across a dark, textured surface. The pieces are in various orientations and positions, some overlapping the boards. The boards show different stages of Tetris games, with some pieces already placed on the grid.

Placing patches

Strategy = Placement policy + Evaluation

Placement policies



- Classic packing heuristics such as Bottom-Left
- CP heuristics based
- Meta-policy for all rotations of patch
- All placements

Which placement?



- Evaluation chooses between alternatives
- Goal is to make best choice
 - First-fail is the wrong approach
 - In essence: only left-most branch of search tree
- Ideas: Left/bottom-most, least bounding box, first, random, ...

Propagation Guided Global Regret

- Choose the placement that makes as little propagation as possible
- Morally inverse of impact based search
- Mathy expression

$$\text{pggr}(B, B', p) = \sum_{i=0}^8 \sum_{j=0}^8 \begin{cases} 0 & \text{if } B'_{ij} = p \\ |B_{ij}| - |B'_{ij}| & \text{otherwise} \end{cases}$$

Which strategy is best?



- 1000 packings for all 119 combinations tested
- Most important thing is propagation guided global regret
- Smart CP heuristics are the wrong choice for policy
- Policy is a choice between speed and quality

Key takeaways



- Game state representation using CP
- Propagation gives us many important signals
- Play Patchwork!
- github.com/zayenz/cp-mod-ref-2019-patchwork

Teaser: Nmbr9 poster tomorrow

- Another fun game!
 - More complicated polyomino placement problem
- Key points
 - Too hard to solve with my current best model
 - Open challenge, do you have a smart idea?
- github.com/zayenz/cp-2019-nmbr9/

Thank you!

~

Questions?