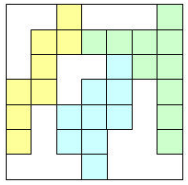
How to decompose a binary matrix into three hv-convex polyominoes

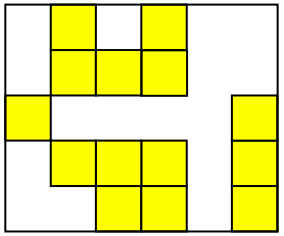
A. Frosini and C. Picouleau

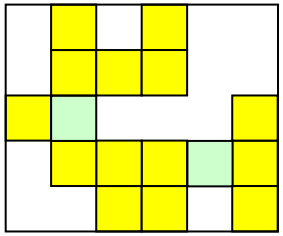


Definitions

polyomino

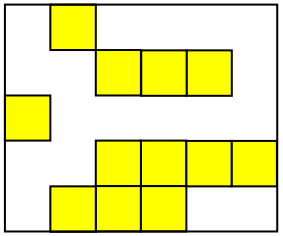
A discrete set of points whose elements are connected (here represented as a set of cells on a squared surface)

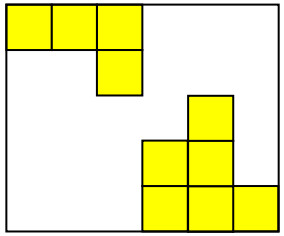




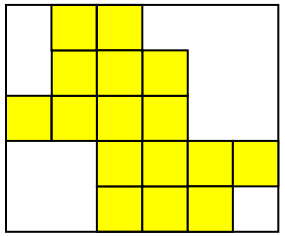
hv-convex set

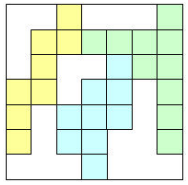
A discrete set of points whose elements are horizontally and vertically convex





polyomino + hv-convex set = hv-polyomino





Definition of the problem and previous results

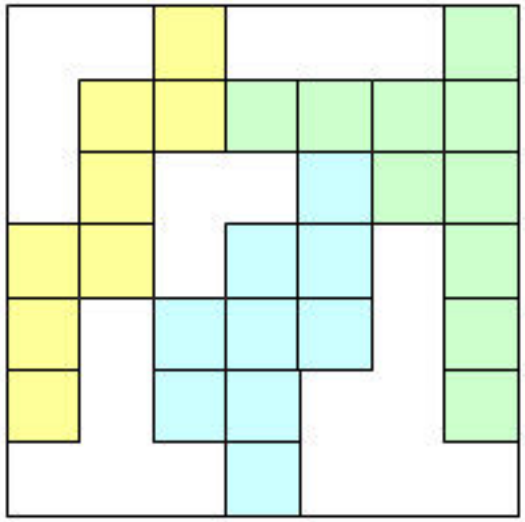
***n*-decomposition**:
given a binary matrix, we want to efficiently decompose it into at most *n hv*-polyominoes, if possible, otherwise give failure.

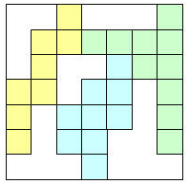
Previous results:
-deciding if a binary matrix can be decomposed into at most *n h*-convex matrices is *NP-complete* (reduction from *3-Partition*)

- the problem *3*-decomposition with respect to *hv*-convex matrices is *NP-complete* (reduction from *3-vertex-coloring*)

- the problem *2*-decomposition both with respect to *hv*-convex matrices and *hv*-polyominoes can decided in *polynomial time*

***3*-decomposition**:
given a binary matrix, we want to efficiently decompose it into at most *3 hv*-polyominoes, if possible, otherwise give failure.

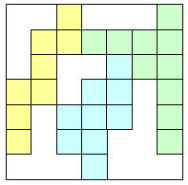




Polynomial time algorithm for *3-decomposition*

Input: a binary matrix A
Output: a decomposition of A into at most 3 *hv*-polyominoes, if possible, otherwise FAILURE

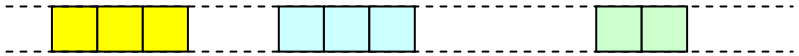
Strategy:
mark with *3* labels, say x, y, and z, the elements of A according to their belonging to *3* polyominoes X, Y, and Z that may constitute the final solution.
The labels are assigned starting from those elements of A where no ambiguities are allowed.

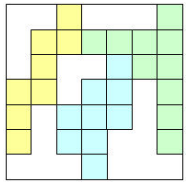


Polynomial time algorithm for *3-decomposition*

Input: a binary matrix A
Output: a decomposition of A into at most 3 *hv*-polyominoes, if possible, otherwise FAILURE

Step 1: perform a preprocessing to avoid trivial cases
Step 2: start the labeling with the elements that lie on rows having two holes

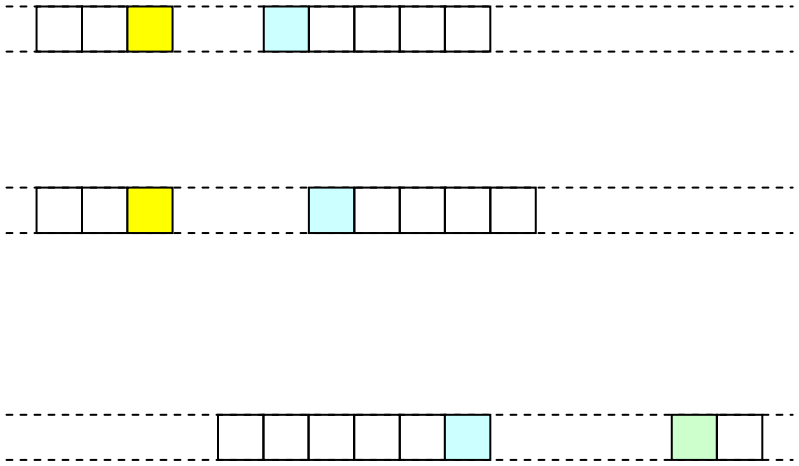


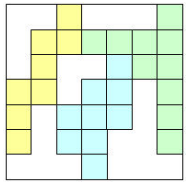


Polynomial time algorithm for *3-decomposition*

Input: a binary matrix A
Output: a decomposition of A into at most 3 hv-polyominoes, if possible, otherwise FAILURE

Step 1: perform a preprocessing to avoid trivial cases
Step 2: start the labeling with the elements that lie on rows having two holes
Step 3: proceed with labeling the rows having one single hole (regarding each hole as a point of a permutation matrix)

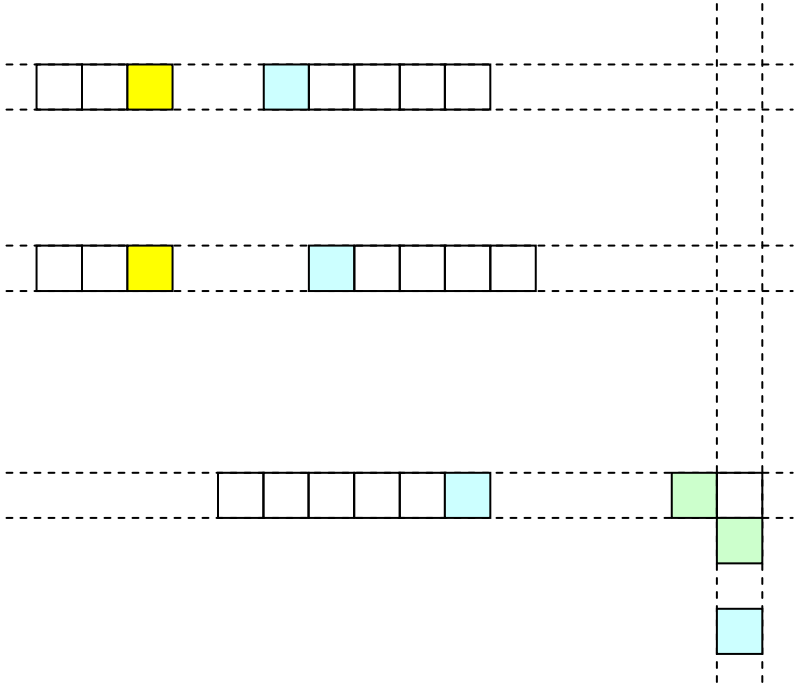


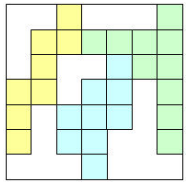


Polynomial time algorithm for *3-decomposition*

Input: a binary matrix A
Output: a decomposition of A into at most 3 *hv*-polyominoes, if possible, otherwise FAILURE

Step 1: perform a preprocessing to avoid trivial cases
Step 2: start the labeling with the elements that lie on rows having two holes
Step 3: proceed with labeling the rows having one single hole
Step 4: label the columns having one single hole in the external areas

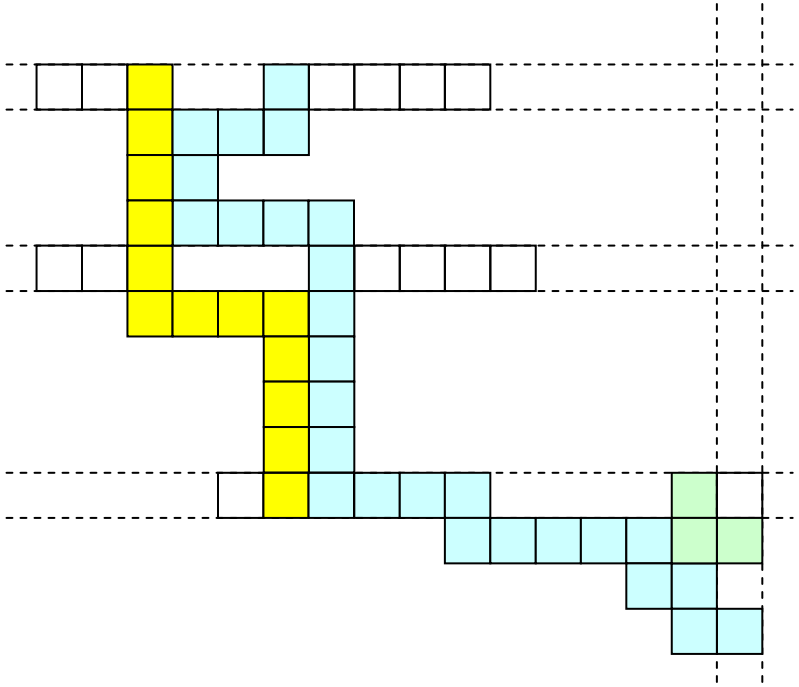


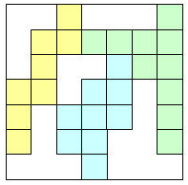


Polynomial time algorithm for *3-decomposition*

Input: a binary matrix A
Output: a decomposition of A into at most 3 *hv*-polyominoes, if possible, otherwise FAILURE

Step 1: perform a preprocessing to avoid trivial cases
Step 2: start the labeling with the elements that lie on rows having two holes
Step 3: proceed with labeling the rows having one single hole
Step 4: label the columns having one single hole
Step 5: complete the borders of the three polyominoes in order to maintain the *hv*-convexity, if possible, otherwise give FAILURE





Three open problems

- find a polynomial time generalization of the algorithm for the decomposition into k different hv-polyominoes;

- motivated from practical problems in Intensity Modulated Radiation Therapy (IMRT), extend the algorithm to the three-dimensional case;

- explore the connections between our problem and the two-sided permutation matrices in order to simplify the algorithm and improve its efficiency.