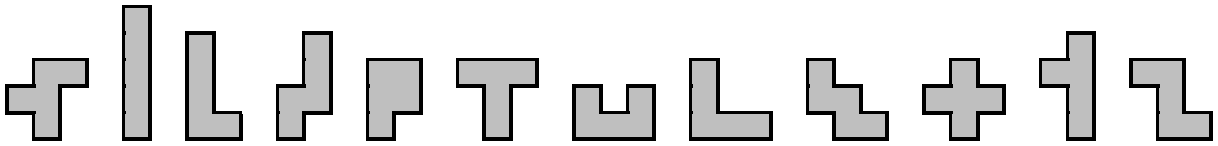
Solution to Puzzle Corner M/J 2

Michael S. Branicky, Course 6, G'96

18 April 2020

The objective is to pick a pentomino tile so as to maximize the number of squares that can be covered with copies of that tile on an 8×8 board without overlap. More generally, for each pentomino tile shape, we will find the maximum number of tiles that can be placed.

We will name the twelve tiles, respectively, F, I, L, N, P, T, U, V, W, X, Y, and Z [1]:

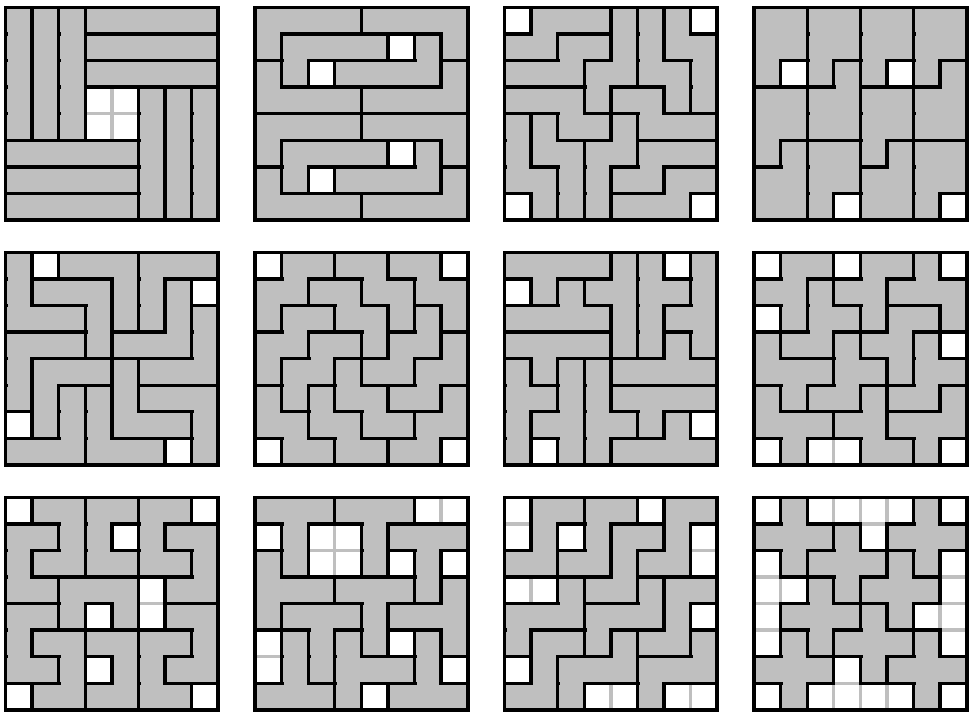


Since each pentomino consists of five squares, the optimal number of tiles that can ever be placed, is $\lfloor 64/5 \rfloor = 12$.

Results

Optimal coverings can be achieved by placing twelve copies of the I-tile, L-tile, N-tile, P-tile, V-tile, W-tile, and Y-tile. The sorted maximal coverings for each pentomino are as follows.

pentomino shape	I	L	N	P	V	W	Y	F	U	T	Z	X
max. tiles placed	12	12	12	12	12	12	12	11	11	10	10	8



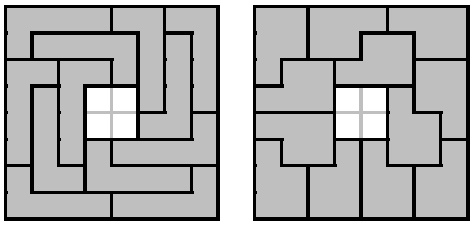
Methods

Some solutions could be found, or tuned, by hand. However, the solutions above and verification of their maximality came from computer search.

All problems were solved by exhaustive search, using backtracking combined with transposition tables (i.e., caching of results) [2, 3]. At each point, the moves considered consisted of every placement of the given tile plus all its (unique) reflections and rotations. However, moves were limited to be within the currently filled window in the board plus a "shell" expanded by the tile's size. Also, the program pruned the search for any board positions that could not accommodate the placement of enough tiles to reach the maximum placed so far.[1]

A separate, exact set cover formulation [4] confirmed the pentomino shapes where twelve tiles could be placed or not, for all tiles but T.[2] Specifically, the set to be covered contains the sixty-four locations on the board plus the use of exactly four monominoes (1×1 squares) representing the locations of the blank cells. The subsets to consider are (i) each of the board positions covered by the given pentomino, ranging over its placements on the board, for itself and for each of its rotations and reflections, and (ii) all sixty-four positions for each of the monominoes. Knuth's dancing links algorithm (DLX), with the S-heuristic, was then used [5]. Eleven-tile solutions for the F-tile and U-tile (and non-solution for X) were confirmed using the same approach plus nine monominoes.[3] This approach was not amenable to finding positive confirmations for the T, Z, and X solutions.

When solved with a 2×2 tetromino instead of four monominoes, exact cover finds optimal coverings for the I-tile (as already shown), plus the L- and P-tiles:



References

[1] `https://en.wikipedia.org/wiki/Pentomino`

[2] `https://en.wikipedia.org/wiki/Backtracking`

[3] `https://en.wikipedia.org/wiki/Transposition_table`

[4] `https://en.wikipedia.org/wiki/Exact_cover`

[5] Knuth, Donald E. Dancing Links. Millenial Perspectives in Computer Science, 2000, 187–214. Available at: `https://arxiv.org/abs/cs/0011047`

[1] Search times ranged from less than 1 minute (I, P, X) through 20-80 minutes (L,T,V,W,Y,Z) to many hours (F,N,U). All times are for Python 3 run in Google Colaboratory.

[2] F was confirmed no, and I, L, N, P were confirmed yes in less than 1s; V and W were confirmed yes, U and X no in about 6s; Y yes, Z no in about 9s. The T search was abandoned after several hours.

[3] F yes in 2 mins, U yes in 23 mins, X no in 27 mins. T and Z searches were abandoned.