Model: 2D Shortest-SuperTile via relative-placement spanning tree

Index sets and parameters

$$
\begin{aligned}
V &= \{1, \dots, T\} && \text{(set of tiles)}\\
\mathcal{I} &= \{1, \dots, n\} && \text{(rows inside a tile)}\\
\mathcal{J} &= \{1, \dots, n\} && \text{(cols inside a tile)}\\
\mathcal{R} &= \{(i, j, k) \;:\; 1 \le i, j \le n+1,\ 1 \le k \le 4\} && \text{(allowed relative placements)}\\
t_{v,a,b} &\in \{0, 1\} \quad \text{(or general symbol)} && \forall v \in V,\ a \in \mathcal{I}, b \in \mathcal{J}\\
\Delta_x(r),\ \Delta_y(r) &\in \mathbb{Z} \quad \text{(offsets implied by relative placement } r \in \mathcal{R}\text{).}
\end{aligned}
$$

Remark: The mapping $r = (i, j, k) \mapsto (\Delta_x, \Delta_y)$ is defined by your corner/offset convention (e.g. corner k of tile v placed into grid relative slot (i, j) of tile u). Use whichever integer offsets you defined in your implementation.

Decision variables

$$
\begin{aligned}
m &\in \mathbb{Z}_+ && \text{(side length of the square canva}\\
x_v, y_v &\in \mathbb{Z} \qquad \forall v \in V && \text{(absolute integer coordinates of the top-left corner of tile } v \text{ on the canva}\\
e_{uvr} &\in \{0, 1\} \quad \forall u, v \in V,\ u \ne v,\ r \in \mathcal{R} && \text{(1 if the tree selects a relative placement } r \text{ that places } v \text{ w.r.t.}\\
\pi_{uv} &\in \{0, 1\} \qquad \forall u \ne v && \text{(undirected tree edge indicator; can be derived from } e_{uvr}
\end{aligned}
$$

Objective

Minimize the canvas size:

$$\min\ m$$

Feasibility / domain constraints

$$0 \le x_v \le m - n, \qquad 0 \le y_v \le m - n \qquad \forall v \in V$$

(tiles must lie wholly inside the $m \times m$ canvas).

Tree / connectivity constraints

We enforce that the chosen relative placements form a spanning tree on V.

$$
\begin{aligned}
&\sum_{u \in V \setminus \{v\}} \sum_{r \in \mathcal{R}} e_{uvr} = 1 && \forall v \in V \setminus \{\text{root}\}\\
&&& \text{(each non-root has exactly one parent (arborescence))}\\
&\sum_{v \in V} \sum_{u \in V \setminus \{v\}} \sum_{r \in \mathcal{R}} e_{uvr} = T - 1 && \text{(total edges} = T - 1\text{)}
\end{aligned}
$$

(Choose an arbitrary root to break symmetry; above enforces an oriented spanning tree.)

Position propagation (consistency) along chosen edges

If $e_{uvr} = 1$ then the absolute positions must satisfy the relative offset implied by r:

$$e_{uvr} = 1 \quad \Longrightarrow \quad \begin{cases} x_v = x_u + \Delta_x(r), \\ y_v = y_u + \Delta_y(r). \end{cases} \qquad \text{(position propagation)}$$

(These are logical implications; linearization suggestions are below.)

Global pairwise compatibility constraints

For every pair of distinct tiles $p, q \in V$ we must ensure that whenever they overlap on the canvas, the overlapped cells are compatible (i.e., have no conflicting symbols). Let

$$\mathcal{O}_{(a,b),(c,d)}(p,q) \;:=\; \text{the event } \big(x_p+(a-1) = x_q+(c-1) \wedge y_p+(b-1) = y_q+(d-1)\big).$$

Then, for every $p \neq q$ and every pair of internal cell indices $(a,b) \in \mathcal{I} \times \mathcal{J}$, $(c,d) \in \mathcal{I} \times \mathcal{J}$ we require:

$$\text{if } \mathcal{O}_{(a,b),(c,d)}(p,q) \text{ holds, then } t_{p,a,b} = t_{q,c,d}. \qquad \text{(compatibility)}$$

Equivalently (compact): for all $p \neq q$,

$$\big(x_p - x_q,\ y_p - y_q\big) = (a - c,\ b - d) \implies t_{p,a,b} = t_{q,c,d}.$$

Alternative formulation using precomputed pairwise compatibility

If you precompute whether two tiles p, q are pairwise compatible under a displacement (δ_x, δ_y), define the binary parameter

$$\mathrm{Comp}_{pq}^{\delta_x,\delta_y} = \begin{cases} 1 & \text{if } \forall (a,b),(c,d) \text{ with } a + \delta_x = c,\ b + \delta_y = d \text{ we have } t_{p,a,b} = t_{q,c,d}, \\ 0 & \text{otherwise.} \end{cases}$$

Then impose for all $p \neq q$:

$$\mathrm{Comp}_{pq}^{x_p - x_q,\ y_p - y_q} = 1.$$

(Again this is a logical constraint; it says the relative offset produced by (x_p, y_p) must be one of the compatible offsets.)

Remarks on implementability and linearization

- The model above uses logical implications (if $e_{uvr} = 1$ then equalities on x, y; and equality tests like $x_p - x_q = \delta_x$ trigger compatibility). To obtain an MILP, linearize as follows:
  - Replace each implication $e_{uvr} = 1 \Rightarrow x_v = x_u + \Delta_x(r)$ with big-M constraints:

    $$x_v - x_u - \Delta_x(r) \leq M\,(1 - e_{uvr}), \qquad x_v - x_u - \Delta_x(r) \geq -M\,(1 - e_{uvr}),$$

    where M is a sufficiently large constant (e.g. $M = m$).
  - For compatibility: for every possible displacement (δ_x, δ_y) and for every ordered pair (p, q), introduce binary indicator d_{pq}^{δ} that equals 1 iff $(x_p - x_q, y_p - y_q) = (\delta_x, \delta_y)$. Enforce one-of constraints and link to the precomputed $\mathrm{Comp}_{pq}^{\delta}$:

    $$d_{pq}^{\delta} = 1 \Rightarrow \mathrm{Comp}_{pq}^{\delta} = 1.$$

    Those implications again are linearizable via big-M.
- The number of possible offsets (δ_x, δ_y) is bounded: since all tiles must lie inside an $m \times m$ canvas, $|\delta_x|, |\delta_y| \leq m - n$. Precomputing $\mathrm{Comp}_{pq}^{\delta}$ for all feasible δ reduces the on-line overlap checking cost.
- Enforcing a spanning tree (acyclic + connected) can be done with standard directed tree/arborescence constraints (e.g. parent variables plus subtour elimination or flow constraints) if you prefer connectivity instead of the simple 'one parent per node' orientation.

Summary (informal)

- Decision variables place each tile at integer coordinates (x_v, y_v) inside an $m \times m$ canvas.
- A subset of relative placements e_{uvr} form a spanning tree; chosen edges force relative offsets between tiles and thus propagate absolute positions.
- Global compatibility requires that any two tiles that end up overlapping have equal symbols in all overlapped cells.
- Objective is to minimize m.