The boundary cut tire T_∂: closing the coverage gap

Motivation

The high-side cut tire forest (`cut_tire_tree_structure.tex`) omits the low-side face of H_1 — the unique face containing the pendants of G'_i. This omission is essential to prove the forest's tree structure (low-side faces span multiple parent faces of H_{d-1}, violating the uniqueness step in the proof).

The coverage gap. Empirically (`chain_dp_joint.py` on the dodecahedron, cut #0, side 0): when $|S_i|$ is small, H_1 on side i can be a *tree* (no cycles). Its unique face is forced to be low-side (contains pendants), so the high-side forest is *empty*. The chain DP, projecting through roots that don't exist, gives $\mathcal{R}_i = \emptyset$ — but G may still be 3-edge-colourable, with non-empty $\mathcal{R}_i$.

Resolution. Introduce the *boundary cut tire* $T_\partial^{(i)}$ on each side i, representing the low-side face of H_1 together with the depth-0 pendants. It is not a child of any other tire (it sits at the framework's outer boundary), but the chain DP can use it as a special root that interfaces between the high-side forest interior and the cut.

Definition

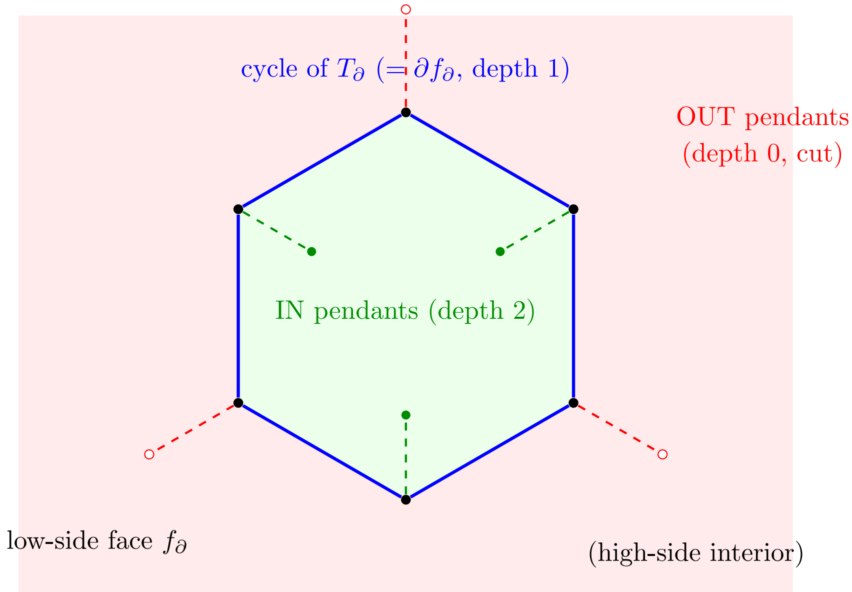
Setup. Fix a side i of a 6-edge cut of G. Let $G'_i = (G[S_i] \cup \text{pendants})$ with edge depths assigned by BFS from pendants (depth 0). H_d = the subgraph of G'_i spanned by depth-d edges.

The low-side face of H_1. By the level-set lemma, each face of H_1 is entirely low-side (interior contains only depth-0 edges = pendants) or entirely high-side (interior contains only depth-≥ 2 edges). There is exactly one low-side face: the unique face whose interior contains the pendants. Call it $f_\partial^{(i)}$.

Definition (Boundary cut tire $T_\partial^{(i)}$)**.** *The* boundary cut tire *on side i is the labelled multigraph $T_\partial^{(i)}$ obtained from:*

- *The boundary walk of $f_\partial^{(i)}$ in H_1 (the edges of H_1 on the boundary of the low-side face), with each depth-*1 *edge contributing one cycle edge.*
- *For each boundary vertex v of $f_\partial^{(i)}$ with* $\deg_{H_1}(v) = 2$*:*
  - *The pendant edge at v (= depth-*0 *cut edge), labelled "OUT."*
  - *If the third edge at v has depth* 2 *(= edge of H_2 in an adjacent high-side H_1 face), one labelled "IN" pendant.*

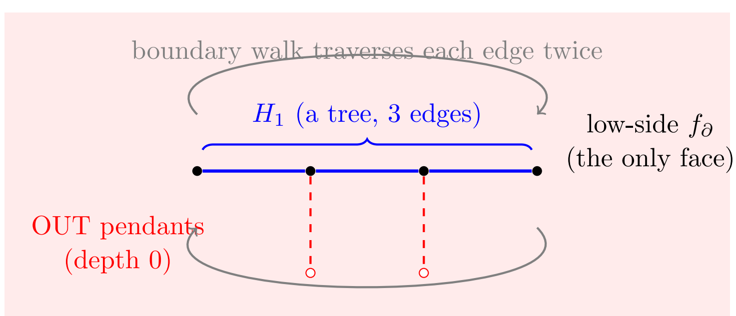
Picture. A thick-side example with H_1 a hexagonal cycle in G_i': the low-side face f_∂ is the outer (unbounded) region containing 3 pendants (depth 0); the adjacent high-side face is inside the hexagon, containing depth-2 edges.



The cycle of T_∂ is the hexagon (blue, depth 1); pendants fall in two classes: OUT (red, dashed, going into the low-side face) attach to the cut, and IN (green, dashed, going into the adjacent high-side region) attach to depth-2 edges shared with $T_2^{(f')}$ tires.

What T_∂ looks like in special cases.

- *Side with thick BFS:* H_1 has multiple faces, one low-side (f_∂) and one or more high-side. T_∂ has cycle = boundary of f_∂ + OUT pendants (= cut edges incident to f_∂'s boundary) + IN pendants (= depth-2 edges in adjacent high-side faces).
- *Side with thin BFS* (e.g. H_1 a tree): The unique H_1 face is f_∂. T_∂ has cycle = boundary walk of this single face (each H_1 edge contributes *two* cycle-edge traversals, since the walk goes around each edge on both sides) + OUT pendants at $V_{\deg=2}$. No IN pendants (no depth-2 edges exist).



In the thin case (e.g. dodecahedron cut #0 side 0 with $|S_0| = 4$), the boundary walk of the single face has length 6 (each of 3 H_1 edges visited twice), and OUT pendants attach at the two $V_{\deg=2}$ vertices.

The extended forest

The high-side cut tire forest of G'_i (proven in `cut_tire_tree_structure.tex`) has roots $T_1^{(f_{\text{high}})}$ for the high-side faces of H_1.

Definition (Extended cut tire structure)**.** *The* extended cut tire structure *on side i is the high-side forest with $T_\partial^{(i)}$ adjoined as a* boundary node*. T_∂ is not a child or parent of any high-side cut tire in the geometric containment sense, but it* shares edges *with adjacent high-side tires:*

- *T_∂ shares depth-1 edges with each high-side depth-1 tire $T_1^{(f_{high})}$ whose face f_{high} is adjacent to f_∂ (= shares a boundary edge of H_1).*
- *T_∂'s IN pendants are cycle edges of depth-2 cut tires $T_2^{(f')}$ in adjacent high-side H_1 faces.*

Role in the chain DP. In the joint-projection chain DP, the per-tire state at T_∂ is a set of valid edge 3-colorings of T_∂'s structure (cycle + OUT + IN pendants). Composition with adjacent tires is via shared edges (same G'_i edge tuple appearing in both), exactly as for the high-side forest. The chain DP then has T_∂ as its *boundary interface*: the OUT pendants project to the cut, and the IN pendants + cycle edges propagate constraints to/from the high-side forest.

Chain DP with T_∂, sketched

1. Process the high-side forest bottom-up as before, producing per-tire valid coloring sets $A(T)$.
2. Compute the valid coloring set $A(T_\partial^{(i)})$ via enumeration of proper 3-edge-colorings on T_∂'s structure.
3. Restrict $A(T_\partial)$ by edge-sharing with adjacent high-side tires:
   - For each $T_1^{(f_{\text{high}})}$ sharing depth-1 edges with T_∂: keep T_∂ colorings consistent with some $A(T_1^{(f_{\text{high}})})$ coloring on shared edges.
   - For each $T_2^{(f')}$ sharing a depth-2 edge with T_∂'s IN pendants: keep T_∂ colorings consistent with some $A(T_2^{(f')})$ coloring on the shared edge.
4. Project $A(T_\partial^{(i)})$ to OUT pendants $= \mathcal{R}_i$.

Where things stand

What this closes. The coverage gap (no high-side cut tires on thin sides) is resolved: T_∂ always exists and provides the interface to the cut.

What it leaves open. T_∂'s "cycle" is the boundary walk of f_∂, which is a closed walk in H_1 — *not* necessarily a simple cycle when H_1 is a tree (the walk traverses each edge twice). The per-tire half (Prop 1.13) does not directly apply to such walks. This is a special case of the "branched cut tires" open problem in `chain_half_analysis.tex`.

Why the chain DP can still work. Even without the per-tire half guaranteeing $|\pi(T_\partial)| \geq 6$, we can *compute* the valid coloring set for T_∂ directly (brute-force or constraint-propagated enumeration). If this set is non-empty after edge-sharing restrictions, the chain DP yields a non-empty $\mathcal{R}_i$.

Logical status of the extended framework.

- Edge-sharing chain DP S_3-equivariance: still holds (uniform S_3 action commutes with edge constraints).
- Tree structure: the high-side forest still forms a forest; T_∂ adjoined is no longer a tree but a connected structure (forest + boundary node).
- Per-tire half for T_∂: open, special case of branched tires.
- Non-emptiness of $\mathcal{R}_i$ via the DP: open; this is the *primary* chain-half claim, now testable with T_∂ included.

Empirical next step

Re-run `chain_dp_joint.py` with T_∂ added. Compare with ground truth (brute-force G'_i 3-edge colorings).

For the dodecahedron cut #0 side 0:

- Ground truth: $|\mathcal{R}_{\text{ground}}| = 36$.
- Old high-side-only DP: $|\mathcal{R}_0| = 0$ (no high-side tires).
- New T_∂-extended DP: should match ground truth (or reveal another gap to investigate).