

Short-Cycle Decompositions and Full Zombie Damage in Cubic Graphs

Lennart Rudolph 

9 September 2026

Abstract

Every connected union H of triangles and quadrilaterals with minimum degree two and maximum degree three has at most four degree-two vertices, called ports. We classify these unions when the number of ports is zero or at least two, including the known closed cubic families. Explicit local strategies and an interface-composition theorem yield a characterization of full zombie damage in finite, simple, connected, bridgeless cubic graphs: the only exceptions are K_4 , $K_{3,3}$, the triangular prism, and the cube. This proves Conjecture 24 of Davila. In every positive case the survivor can force full damage within $2n(n+1)$ moves, against every choice among geodesic zombie replies. If the graph has an edge on neither a triangle nor a quadrilateral, the survivor can safely depart from every vertex infinitely often. Five finite routing components are supported by complete ranked policies with 19 representative tasks and 121 states. An end-to-end Lean 4.32.0 formalization proves the universal classification, four-port bound, full-damage characterization and recurrent coverage, including the construction of the component models, routing strategies and initial placement from the graph hypotheses.

Keywords: short-cycle decomposition; cubic graph; zombie damage; strategy certificate.

2020 Mathematics Subject Classification: 05C75, 05C57, 05C12.

1 Introduction and results

Local cycles can create alternative shortest paths and obstruct the composition of simple traversal rules. In graphs of maximum degree three, however, connected regions made from triangles and quadrilaterals have a uniformly bounded interface. Our approach is to isolate these regions, describe their available boundary behavior, and then compose verified local strategies on the graph connecting them. This separates a structural statement about graphs from its pursuit-game application.

Graphs are finite and simple; ambient graphs are connected unless stated otherwise. The explicitly defined quotient is a multigraph, retaining loops and parallel edges. A graph is *subcubic* if its maximum degree is at most three. A *short cycle* means a triangle or quadrilateral. For a connected nontrivial union H of short cycles in a subcubic graph, every internal degree is two or three. Define its unused degree capacity by

$$p(H) = \sum_{v \in V(H)} (3 - \deg_H(v)).$$

Thus $p(H)$ counts the degree-two vertices, called *ports*.

Theorem 1 (Four-port principle). *Let H be a finite, simple, connected graph with minimum degree two, maximum degree at most three, and every edge on a triangle or quadrilateral in H . Then $p(H) \leq 4$. In any subcubic supergraph containing H , at most four edges leave $V(H)$.*

The proof and the complete component classification appear in Section 3. The main invariant is elementary: attaching a new short cycle never increases $p(H)$. Long components are ladders or capped ladders; the remaining open cases form a finite list. The boundary bound also gives the expansion consequence in Corollary 8. The interface-composition theorem, Theorem 13, is not restricted to degree three or to a particular pursuit game: its hypotheses state precisely the safety and finite-completion guarantees that local strategies must provide.

Relation to earlier structure theory. The closed cubic case is already known. Cushing, Kangaslampi, Lipiäinen, Liu, and Stagg classify cubic graphs with nonnegative zero-idleness Ollivier–Ricci curvature as prisms and Möbius ladders, including the small cases [8, Lemma 5.1 and Corollary 5.4]. The lemma numbering refers to arXiv:1712.03033v2 (22 December 2017); the published version appeared in *Experimental Mathematics* in 2022. Their edge-curvature criterion identifies that class with the cubic graphs in which every edge is on a triangle or square. We include an elementary attachment proof to obtain the open-component interfaces needed here; the closed classification is not claimed as new. The curvature results are not premises of the routing proof.

Application to zombie damage. In the one-zombie damage game, the zombie chooses its starting vertex, the survivor chooses a starting vertex after observing it, and the zombie moves first. Each zombie move must reduce its distance to the survivor by one. The survivor may traverse an edge or pass. A vertex is damaged when the survivor survives a zombie turn there and completes a survivor move from it; a pass counts, and repeated visits do not increase the score. The zombie minimizes and the survivor maximizes the number of distinct damaged vertices. Randy Davila introduced this value as the *zombie damage number* $\text{zdmg}(G)$ [1, Definition 1].

Put $n = |V(G)|$. A graph has *full zombie damage* when $\text{zdmg}(G) = n$. Davila proves this for graphs of minimum degree at least two and girth at least five [1, Theorem 15]; Corollary 16 is its cubic specialization. His Conjecture 24 asks whether every bridgeless cubic graph of order at least ten has full damage. The application below gives the complete four-exception characterization.

Related pursuit and damage games. Cox and Sanaei introduced the ordinary damage number [2]. Subsequent work includes Cartesian products by Huggan, Messinger, and Porter [3, 4], multiple robbers by Stojaković and Wulf [5], and further graph families and complexity results by Gledel, Kinnersley, Patkós, and Stojaković [6]. These games allow an unrestricted cop to choose a neighboring vertex or pass. Liu and Miller study when geodesic zombies suffice for capture and bound capture throttling on grids [7]. Their objective differs from the damaged-vertex payoff used here. Davila’s parameter combines this payoff with the geodesic pursuer restriction; our application concerns that combination.

Write $P_m = C_m \square K_2$ for the prism, $m \geq 3$, and M_{2m} for the Möbius ladder with a $2m$ -cycle rim and edges joining antipodal vertices. Thus $M_6 = K_{3,3}$, while M_8 is not the cube. Write Q_3 for the cube.

Theorem 2 (Full-damage application). *Let G be a finite, simple, connected, bridgeless cubic graph. Then*

$$\text{zdmg}(G) = |V(G)| \iff G \not\cong K_4, K_{3,3}, P_3, Q_3.$$

In every positive case the survivor can force full damage within $2n(n+1)$ survivor moves, for every initial zombie vertex and every subsequent choice among shortest paths.

Corollary 3 (Davila's Conjecture 24). *Every finite, simple, connected, bridgeless cubic graph of order at least ten has full zombie damage.*

Call an edge *clean* if it belongs to no triangle or quadrilateral. The stronger routing conclusion is as follows.

Theorem 4 (Recurrent coverage). *If a finite, simple, connected, bridgeless cubic graph has a clean edge, then, from every initial zombie vertex, the survivor has a strategy that avoids capture forever and makes safe departures from each vertex infinitely often.*

Repeated safe departures describe recurrence, not a damage score exceeding n . Large prisms require a separate argument that covers each rail once. The local obligations are proved for the ladder families and, for five finite components, by the complete policies in Appendix A. Finally, Remark 19 uses distance-shell observations to distinguish coverage from observability on an explicit infinite family. That comparison uses only the single-target trade obstruction, not the earlier paper's grid theorem.

2 Adjacent positions and clean transitions

We record positions immediately after a zombie move. In a *routing position* the zombie is at u , the survivor is at v , and $uv \in E(G)$. The survivor chooses a neighbor w of v such that

$$w \neq u, \quad uw \notin E(G). \quad (1)$$

Then $d(u, w) = 2$, and the next zombie destinations are precisely

$$R(u, w) = N(u) \cap N(w). \quad (2)$$

Indeed, a neighboring vertex of u reduces the distance from two to one exactly when it is also adjacent to w . The move $v \rightarrow w$ damages v , not w . The set in (2) is nonempty because it contains v .

Lemma 5 (Clean transition). *A two-edge path with distinct endpoints that contains a clean edge is induced and is the unique two-edge path between its endpoints. In particular, if the survivor crosses a clean edge vw from a routing position (u, v) with $u \neq w$, its move is safe and the next zombie move is forced to v .*

Proof. An edge joining the endpoints would give a triangle containing the clean edge. A second common neighbor of the endpoints would give a quadrilateral containing that edge. Both are excluded. The last assertion follows from (2). $\square$

An *entry through a clean edge* xp is the routing position (x, p) . A safe exit across a clean edge automatically becomes another entry after the forced zombie move. This transition is the interface used below; it remains valid when both endpoints of the clean edge lie in the same short-cycle component.

3 Subcubic short-cycle structure

In this section G may be subcubic. Let Q be the spanning subgraph of G whose edges are exactly the edges belonging to a triangle or quadrilateral. We do not assume that Q , or any of its components, is induced. Every vertex of a nontrivial component H has degree two or three in H , because each of its incident Q -edges belongs to a cycle. A degree-two vertex of H is a *port*. It has at most one incident clean edge in G , and exactly one when G is cubic. An isolated vertex of Q has only clean incident edges; there are three in the cubic case.

3.1 The families

For $m \geq 2$, let L_m have vertices $a_1, \dots, a_m, b_1, \dots, b_m$, rail edges $a_i a_{i+1}, b_i b_{i+1}$, and rungs $a_i b_i$. Its four ports are a_1, b_1, a_m, b_m . For the following definitions only, take L_1 to be the edge $a_1 b_1$.

Let T_m , $m \geq 1$, be L_m with a new vertex c joined to a_1, b_1 . Its ports are c, a_m, b_m , and T_1 is a triangle. Let D_m , $m \geq 1$, add also a second cap d joined to a_m, b_m . Its ports are c, d , and $D_1 = K_4 - e$ is the diamond.

Five further open components occur:

$$K_{2,3}, \quad K_{3,3} - e, \quad P_3 - e_\Delta, \quad Q_3 - v, \quad Q_3 - e. \quad (3)$$

Here e_Δ is a triangle edge of the triangular prism, not a matching edge. The port counts in (3) are respectively 3, 2, 2, 3, 2.

Theorem 6 (Short-cycle classification). *Every nontrivial component of Q is one of the families L_m, T_m, D_m , one of the five graphs in (3), a graph with exactly one port, or a closed graph from the list*

$$K_4, \quad K_{3,3}, \quad P_m \ (m \geq 3), \quad M_{2m} \ (m \geq 3). \quad (4)$$

Here “closed” means having no ports. Repetitions in the list are harmless.

Proof. Start with one short cycle and adjoin the remaining short cycles in a connected order. Here two cycles are adjacent in the cycle-intersection graph *exactly when they share an edge*. This graph is connected: connectedness of the union first gives connectivity through shared vertices, and two cycles meeting at a vertex in a subcubic graph share an incident edge, since otherwise that vertex has degree at least four.

At every stage the current union has minimum degree two and maximum degree three. Consider a newly adjoined cycle that contributes an edge. Its intersection with the union has no isolated vertex: such a vertex would receive two new incident edges in addition to at least two old ones. The nonempty intersection is consequently a union of edge-paths. A cycle of length at most four allows only the following attachments.

Type A. Join two distinct old ports by a path with r new edges and $r - 1$ new internal vertices. A simple old path of length $\ell \geq 1$ completes the new cycle, where

$$\ell + r \in \{3, 4\}, \quad r \geq 1.$$

A new edge must not duplicate an old edge. Two old ports are consumed and $r - 1$ new ports are created, so the port count changes by $r - 3 \in \{0, -1, -2\}$.

Type B. The intersection consists of two disjoint old edges. All four endpoints are ports, and the two new edges complete a quadrilateral. Four ports are consumed.

Thus the port count never increases from its initial value, three or four. This already proves the first assertion of Theorem 1. In a subcubic supergraph every edge leaving $V(H)$ consumes one of

these unused incidences, which proves its second assertion. A graph with one port permits neither attachment and is terminal. A graph with no ports is likewise terminal.

For completeness, Table 1 gives the complete Type A and Type B transition lists for unions with at least two ports. In that table $1p$ denotes an unspecified one-port terminal graph. To verify a row, list the port pairs with an old simple path of length at most three. A one-edge attachment is allowed for an old path of length two or three, a two-edge attachment for length one or two, and a three-edge attachment only for length one. Existing edges cannot be added again.

Current union	Type A outcomes	Type B outcomes
L_2	$L_3, T_2, D_1, K_{2,3}$	K_4
L_3	$L_4, T_3, Q_3 - v, P_3 - e_\Delta, K_{3,3} - e$	$P_3, K_{3,3}$
L_4	$L_5, T_4, Q_3 - e$	P_4, M_8
$L_m, m \geq 5$	L_{m+1}, T_m	P_m, M_{2m}
T_1	T_2, D_1	none
T_2	$T_3, D_2, P_3 - e_\Delta, 1p$	none
T_3	$T_4, D_3, 1p$	none
$T_m, m \geq 4$	T_{m+1}, D_m	none
D_1	$K_4, 1p$	none
D_2	P_3	none
$D_m, m \geq 3$	none	none
$K_{2,3}$	$K_{3,3} - e, 1p$	none
$K_{3,3} - e$	$K_{3,3}$	none
$P_3 - e_\Delta$	$P_3, 1p$	none
$Q_3 - v$	$Q_3 - e, 1p$	none
$Q_3 - e$	Q_3	none

Table 1: Complete short-cycle attachment outcomes, up to isomorphism. One-port unions are terminal. Type B consumes four ports and so occurs only on a ladder.

Here are explicit checks that also justify the unbounded rows. With four ports, the union starts as L_2 . Preserving four ports requires a three-edge attachment along an edge joining two ports. In L_2 all choices are equivalent; in L_m for $m \geq 3$ these edges are precisely the end rungs. The result is L_{m+1} .

Reducing four ports to three adds a two-edge path. Along an end rung this gives T_m . The only other eligible port pairs are opposite vertices of L_2 , giving $K_{2,3}$, and same-rail endpoints of L_3 , giving $Q_3 - v$. Reducing four ports directly to two adds a single edge. The nonadjacent eligible pairs give D_1 from L_2 , $P_3 - e_\Delta$ or $K_{3,3} - e$ from L_3 , and $Q_3 - e$ from L_4 . Opposite-end ports of L_m have distance at least $m - 1$, so no such pair is eligible for $m \geq 5$.

With three ports, the initial triangle is T_1 . In T_m a square attached along the open-end rung gives T_{m+1} and a triangle there gives D_m . The cap-to-open-end distances are m . They yield the additional $P_3 - e_\Delta$ and one-port cases in T_2 , and the one-port case in T_3 ; for $m \geq 4$ they are too large. In $K_{2,3}$ and $Q_3 - v$ the ports are independent and every port pair has distance two. Adding a two-edge path gives respectively $K_{3,3} - e$ and $Q_3 - e$, while adding an edge gives a one-port graph.

All resulting two-port graphs have nonadjacent ports. In D_m their distance is $m + 1$: D_1 permits the closed K_4 and a one-port attachment, D_2 closes to P_3 , and D_m for $m \geq 3$ admits no attachment. In $K_{3,3} - e$ and $Q_3 - e$ the ports have distance three, so the only attachment restores the missing edge. In $P_3 - e_\Delta$ the ports have distance two, so restoring the missing edge gives P_3 and adding a two-edge path gives a one-port graph. This verifies the remaining rows.

Finally, Type B can occur only with four ports, hence on a ladder. For $m \geq 3$ its two old edges are the end rungs. Closing them with untwisted or twisted rail edges gives P_m or M_{2m} . In L_2 the

valid closure gives K_4 . The transition list is closed under both attachment types and contains both initial cycles T_1, L_2 . Induction over the adjoining order proves the classification. $\square$

Lemma 7. *If G is cubic and bridgeless, no nontrivial component of Q has one port. A component with no ports is the whole graph. If G has a clean edge, contracting every component of Q and retaining all clean edges yields a connected bridgeless multigraph B .*

Proof. A one-port component would have exactly one edge leaving its vertex set. That edge would be a bridge. A component with no ports is already cubic and has no edge leaving it, so connectedness makes it all of G .

Contraction preserves connectedness. A bridge of B would separate two collections of components joined by only that clean edge in G , making it a bridge of G as well. Clean edges joining ports of the same component become loops; different clean edges with the same component endpoints remain distinct parallel edges. $\square$

Corollary 8 (Expansion obstruction). *Let G be a finite connected subcubic graph and let H be a proper nontrivial short-cycle component. For*

$$\eta(G) = \min_{\emptyset \neq S \subseteq V(G), |S| \leq n/2} \frac{|\partial S|}{|S|},$$

where ∂S consists of edges with one endpoint in S , we have

$$\min\{|V(H)|, n - |V(H)|\} \leq \frac{4}{\eta(G)}.$$

Proof. Theorem 1 gives $|\partial V(H)| \leq 4$. Apply the definition of $\eta(G) > 0$ to the smaller side of this cut. $\square$

The restrictions are substantive. Attaching a pentagon along one old edge adds a path of four new edges and increases the unused degree capacity by one. Repeated such attachments yield arbitrarily many ports in subcubic graphs. In degree four, even squares can meet at a single vertex, invalidating the attachment reduction above.

4 Local routing and the infinite families

Return now to a cubic ambient graph. Fix a nontrivial open component H . To describe its local game, attach a new leaf p^* to each port p . An entry at p is the post-zombie position (p^*, p) . Exiting at q means moving from q to q^* . The leaves are a local notation, not a claim that the ambient endpoints are leaves or distinct vertices.

We require two properties, always with finite completion and survival:

(A) **Prescribed-exit routing.** For every pair of distinct ports p, q , entry at p permits the survivor to force an exit at q .

(B) **Target routing.** For every entry port p and every vertex $t \in V(H)$, the survivor can damage t and then force an exit through some port other than p . The exit may depend on the zombie's choices.

Lemma 9 (Local-to-ambient transfer). *Every local strategy satisfying (1) against all responses (2) is a valid ambient strategy up to its clean exit. This remains true when a clean edge joins two ports of the same component.*

Proof. For an internal proposed move, the old zombie vertex u and the survivor vertices v, w are in H , with $uv, vw \in E(H)$. An additional ambient edge uw outside Q would complete a triangle, and hence would itself be in Q . An additional ambient common neighbor x producing a second two-edge path u, x, w creates a quadrilateral u, v, w, x, u when $x \neq v$. Its edges then belong to Q and to the same component H . Thus safety and the response set for an internal move are computed exactly in H .

For the first move from an entry, the path through the entering clean edge has a unique middle vertex by Lemma 5; that zombie move is forced. The same lemma applies at exit. The exiting clean edge cannot coincide with the internal edge occupied by the zombie.

If p^* denotes another port of H in the ambient graph, the first transition is still forced by the clean edge. Afterwards both occupied vertices are ordinary internal vertices, so the preceding internal argument applies. A premature coincidence along a competing path of length at most two would put the clean edge in a triangle or quadrilateral. Two different stubs may also represent the same external vertex: their entry and exit transitions are still forced, and any competing internal two-edge shortcut was excluded above. No injectivity of the stub notation is required. After exit the forced zombie move gives an entry at the other endpoint, even if it is again in H . $\square$

Lemma 10 (Ladder routing). *Every L_m, T_m, D_m satisfies (A) and (B). Each route can be completed within $2|V(H)|$ survivor moves, counting the clean exit.*

Proof. While the survivor advances monotonically along a rail, two kinds of adjacent positions suffice. In a *trailing position* the zombie occupies the preceding vertex of the same rail. A forward step forces it to follow. In a *rung position* the zombie occupies the opposite vertex of the survivor's current rung. After a forward step it can remain opposite at the new rung, or join the survivor's rail immediately behind. These are all the common neighbors in (2). Once trailing, it remains trailing as long as forward rail motion continues. At an open end, either position permits a clean exit.

Uncapped ladders. By symmetry enter L_m at a_1 . To exit at b_1 , cross the first rung and leave. To exit at a_m , follow the a -rail; to exit at b_m , cross the first rung and follow the b -rail. For a target, choose the rail containing it and use the corresponding route. These paths damage their endpoint on exit, use a nonentry exit, and take at most $m + 1$ moves.

One cap. First let $m \geq 2$. Entry at c permits a choice of either rail followed by its open-end exit. This proves both routing properties for that entry. For entry at a_m , follow the a -rail and the cap to exit at c , or cross the entry rung to exit at b_m ; this proves (A). Targets on the a -rail or at c are handled by the route to c .

For a target on the b -rail, cross from a_m to b_m and move toward b_1 . If the zombie has joined the trailing position, continue to c and exit. Otherwise, at b_1 the zombie is at a_1 . The cap is then blocked, but reversing along the b -rail is safe: from (a_1, b_1) the move to b_2 has only the usual rung/trailing replies. Return along that rail to b_m and leave there. The departure from b_1 damages it even in the reversing case; every other target on the rail was already damaged on the outward journey. The exit is not a_m . The reversed route uses $1 + (m - 1) + (m - 1) + 1 = 2m$ moves. The other routes are no longer. For $m = 1$, all three vertices are ports of a triangle; move directly to the desired other port and exit, covering any port target by this choice.

Two caps. For $m \geq 2$, enter at c , choose a rail containing the target, continue to d , and exit. The zombie is forced to trail and the clean exit is safe. Targets at c and d are covered on departure

as well. This takes $m + 2$ moves. For $m = 1$, use $c \rightarrow a_1 \rightarrow d \rightarrow d^*$ or $c \rightarrow b_1 \rightarrow d \rightarrow d^*$. The zombie may choose either middle vertex after the second survivor move, but neither prevents the clean exit. There is only one nonentry exit, proving both requirements. All displayed bounds are at most $2|V(H)|$. $\square$

5 The five finite routing proofs

The finite cases will be proved by explicit policies, not by asserting the output of a search solver. Label them as follows:

Graph	Vertex and edge specification
$K_{2,3}$	Parts $\{0, 1\}$ and $\{2, 3, 4\}$.
$K_{3,3} - e$	Parts $\{0, 1, 2\}$ and $\{3, 4, 5\}$; delete 03.
$P_3 - e_\Delta$	Triangles 012, 345 and matching 03, 14, 25; delete 01.
$Q_3 - v$	Binary labels $0, \dots, 7$; adjacency differs in one bit; delete vertex 7.
$Q_3 - e$	The same cube; delete edge 01.

A task consists of an entry port p , allowed exit ports E with $p \notin E$, and a target t , which may be absent for (A). A policy state is (b, u, v) , where $b \in \{0, 1\}$ records whether the target has already been damaged. The policy supplies a move w and a positive integer rank ρ . When t is absent the target condition is automatically satisfied after a move. Otherwise the flag after moving from v is

$$b' = b \vee [v = t]. \quad (5)$$

The initial state is $(0, p^*, p)$.

Lemma 11 (Ranked policy soundness). *Suppose a finite table contains the initial state. At every listed state (b, u, v) , require $b \in \{0, 1\}$, $v \in V(H)$, uv to be an edge, a positive integer rank, and a prescribed move $w \in N(v)$ satisfying (1). All edges and neighborhoods here are in the leaf-augmented component. If the move exits at q^* , then $q \in E$ and the updated target flag is true. Otherwise require $w \in V(H)$ and that every state*

$$(b', x, w), \quad x \in N(u) \cap N(w), \quad (6)$$

appears in the table with a strictly smaller positive rank. Then the table is a survivor strategy that completes the task in at most the initial rank many moves.

Proof. Safety prevents capture on the next zombie turn. Formula (2) lists all legal replies, and (5) credits the correct departure vertex. After a nonterminal move every possible reply has a lower rank. Induction on the current positive rank proves finite termination at an allowed exit with the target already damaged. The same induction bounds the number of survivor moves by the current rank. In particular, no assumption is made about which shortest path the zombie selects. $\square$

5.1 Symmetry and task coverage

It suffices to prove (B) for degree-three targets. If the target is the entry port, any (A) route damages it on departure. If it is another port, the (A) route exiting there damages it on its final move.

Table 2 specifies all remaining representative tasks. The first displayed entry port has representatives for every ordered entry/exit pair under automorphisms; the internal target representatives

cover the orbits of its stabilizer. The following permutations are explicit automorphisms under the preceding labels:

$$\begin{aligned}
K_{2,3} &: (01), (234), (34); \\
K_{3,3} - e &: (12), (45), (03)(14)(25); \\
P_3 - e_\Delta &: (01)(34); \\
Q_3 - v &: (12)(56), (24)(35); \\
Q_3 - e &: (24)(35), (01)(23)(45)(67).
\end{aligned}$$

These can be checked directly on the edge lists. For $Q_3 - v$, the generators permute the binary coordinate positions; for $Q_3 - e$, they interchange the two coordinates other than the deleted-edge direction and flip that direction. The displayed groups are transitive on ordered pairs of distinct ports. Their entry stabilizers give exactly the target orbit representatives in the table. Completeness of an automorphism group enumeration is not needed: these specific actions already cover all required tasks.

H	Entry	(A) exit	Internal targets for (B)	Tasks	Rows	Max. rank
$K_{2,3}$	2	3	0	2	8	3
$K_{3,3} - e$	0	3	1, 4	3	18	4
$P_3 - e_\Delta$	0	1	2, 3, 4, 5	5	26	8
$Q_3 - v$	3	5	0, 1, 4	4	28	5
$Q_3 - e$	0	1	2, 3, 6, 7	5	41	10
Total				19	121	

Table 2: Complete representative task coverage. A target task allows every nonentry port as exit.

Lemma 12 (Exceptional-component routing). *Each graph in (3) satisfies (A) and (B), with at most $2|V(H)|$ survivor moves per route.*

Proof. Use the complete tables in Appendix A. For each row, compute the common neighbors of u, w from the displayed graph and port-leaf construction. An internal move has precisely the successor rows (6), all of smaller rank. An exit is through an allowed port and has its target flag true. These finite checks establish the hypotheses of Lemma 11 for every printed task. The symmetry actions above and the port-target observation cover all of (A) and (B). The largest ranks are 3, 4, 8, 5, 10 on graphs of orders 5, 6, 6, 7, 8, respectively, giving the asserted bound. $\square$

The archive contains a separate distance-based checker for these rows: it computes actual graph distances in the leaf-augmented component and checks every neighbor decreasing distance, rather than simply reusing a common-neighbor search routine. It also checks the graph labels, the permutations as edge-preserving maps, and explicit task coverage. These executions supplement the printed finite proof; no output of a strategy synthesizer is itself a premise of Lemma 12.

6 Global composition and initialization

6.1 A degree-independent interface theorem

Let B be a finite connected bridgeless multigraph with at least one edge. Each vertex i represents a module with a finite set W_i of local targets. A port is an incident *edge-end*; the two ends of a

loop are distinct ports. A valid entry records its entering port and whatever local state is needed. Crossing an interface produces a valid entry in the next module. The target sets are disjoint, and there are finitely many targets in total. Move bounds count actions of the controlled player; opponent replies and the interface update after an exit are not counted separately.

For each module fix a positive integer c_i . Assume two contracts hold from *every* valid entry state, against every legal continuation of the opponent. First, any specified different port can be reached and exited within c_i moves. Second, any specified local target can be served and a different port exited within c_i moves. A route stays in its module until its final exit, preserves the stated safety condition throughout, and the interface transition also preserves safety. The exit in the target contract may depend on the opponent. These are semantic requirements on strategies, not just assertions about an underlying undirected path.

Theorem 13 (Interface composition). *Under these contracts, from any valid entry, any prescribed target can be served safely and another valid entry reached within*

$$C = \sum_{i \in V(B)} c_i$$

moves. Any finite list of targets can be served in its prescribed order, and any nonempty finite list can be served repeatedly forever. A list of N targets requires at most NC moves for one pass.

Proof. Suppose the current module is i , the entering edge is e , and the target lies in module j . If $i = j$, use the target contract. Otherwise $B - e$ is connected, even when e is a loop, because B is bridgeless. Choose a simple path from i to j in $B - e$. At each intermediate module use the prescribed-exit contract. The exit port differs from the entry port: the first edge is not e and consecutive edges on the path are distinct. At j use the target contract. Each crossed interface gives a valid entry, so all contracts remain applicable against every opponent response.

The simple path visits each module at most once. Summing its route bounds gives at most C moves, including the final exit. The resulting entry supports the next target request, proving the finite-list assertion by induction. Every route terminates and includes at least its exit move. Repeating a nonempty finite list therefore defines an infinite safe play and serves each listed target infinitely often. Loops and parallel edges retain their edge and port identities throughout. $\square$

This elementary composition argument does not assume a pursuit metric or a degree bound. In the application below, the substantial obligations are proving the local contracts and ensuring that the leaf-augmented model faithfully represents every actual zombie response.

6.2 Application to the short-cycle quotient

Lemma 14 (Routing to a global target). *Suppose G is bridgeless and cubic and has a clean edge. From any clean-edge entry, the survivor can force a safe departure from any prescribed vertex and reach another clean-edge entry. This takes at most $2n$ survivor moves.*

Proof. Use the quotient B of Lemma 7, with each original vertex as a local target in its component. In every nontrivial component, Lemmas 10 and 12 supply the contracts with $c_H = 2|V(H)|$. Lemma 9 proves their ambient validity. At an isolated vertex of Q , any clean edge other than the entering edge is a legal exit; that move serves its sole target and costs one move. Cubicity supplies three possible incident edges. Lemma 5 makes every interface transition safe and produces a valid entry.

Consequently Theorem 13 applies, with $C \leq 2n$, since component vertex sets partition $V(G)$. The port convention also covers a clean edge with both endpoints in one component: its two edge-ends are different ports of a quotient loop. Exiting through the other end traverses that loop without immediately reversing through the entry port. $\square$

Lemma 15 (Initial placement). *For every initial zombie vertex, the survivor can choose a starting vertex and reach a clean-edge entry without capture in at most $2n$ survivor moves.*

Proof. Fix a clean edge ab and an arbitrary starting zombie vertex z . Suppose first that $z \notin \{a, b\}$. At least one endpoint, say a , is not adjacent to z , since otherwise z, a, b would make a triangle. Place the survivor at a . Pass while the zombie is still more than one edge away after its move. Its distance to the stationary survivor decreases on every zombie turn, so it eventually reaches a neighbor u of a without capture. If $u = b$, this already is the entry (b, a) . If $u \neq b$, cross ab ; the next zombie move is forced to a , giving entry (a, b) . There are at most n survivor moves in this case.

Now suppose $z = a$. If b is an isolated vertex of Q , choose a different clean edge br and start the survivor at r . The path a, b, r is the unique two-edge path, so the first zombie move is forced to b and yields entry (b, r) .

Otherwise consider any (A) strategy at the hypothetical entry (a, b) into the component containing b . Such a strategy exists since that component has at least two ports. Let its first survivor move be $b \rightarrow r$. Start the actual survivor at r . Again the clean edge makes a, b, r the unique two-edge path, so the first actual zombie move is forced to b . The position now equals the continuation of the hypothetical strategy immediately after its first move and the corresponding forced zombie reply. Follow the rest of that strategy to its clean exit. Property (A) has no target requirement, so the skipped departure from b creates no unmet obligation. The local bound is at most $2|V(H)| \leq 2n$ moves; skipping its first move only decreases this number. The case $z = b$ is symmetric. $\square$

Proof of Theorem 4. Use Lemma 15 to obtain an entry. Order the finitely many vertices of G and apply Lemma 14 successively to them. At each completion a safe departure from the target has occurred and a new entry is available. Repeat the target list forever. Every local route terminates, so every vertex is departed from safely infinitely often. All moves and all intervening zombie responses are safe. $\square$

The first traversal of the target list uses at most $n(2n)$ survivor moves after initialization. Hence every clean-edge graph has full damage within

$$2n + 2n^2 = 2n(n + 1) \tag{7}$$

survivor moves. This is an upper bound, not an assertion of optimal cover time.

7 The closed families and the four exceptions

If no clean edge exists then $Q = G$. Theorem 6 places G in (4). We now treat this list and explicitly exclude the four small exceptions.

Lemma 16. *The Möbius ladder M_{2m} has full zombie damage for every $m \geq 4$. The survivor can make safe departures from every vertex infinitely often.*

Proof. Index the rim modulo $2m$. The neighborhoods of $i - 2$ and i are

$$\{i - 3, i - 1, i - 2 + m\}, \quad \{i - 1, i + 1, i + m\}.$$

For $m \geq 4$ their only common element is $i - 1$, and the two vertices are nonadjacent. To check the first assertion, compare the other two elements in each set: any unwanted equality would require $m \leq 3$ or a nonzero multiple of $2m$ of absolute value strictly less than $2m$. Rotational symmetry permits the zombie to start at 0; start the survivor at 2 and move forward forever. Each zombie move is forced to follow the rim trace. The first $2m$ survivor departures cover all vertices. $\square$

Lemma 17. *The prism P_m has full zombie damage for every $m \geq 5$, achieved in $2m$ survivor moves.*

Proof. Write its rails as a_i, b_i with indices modulo m , and its rungs as $a_i b_i$. By symmetry the zombie starts at a_0 ; place the survivor at a_2 . Follow the a -rail in the sequence

$$a_2, a_3, \dots, a_{m-1}, a_0, a_1.$$

For $m \geq 5$ each two-edge rail path is the unique geodesic between its endpoints, so the zombie is forced to trail. When the survivor is at a_1 and the zombie has just moved to a_0 , cross to b_1 . This departure completes the first rail's damage.

Continue forward forever on the b -rail. Just before this continuation, the next zombie reply is either a_1 (opposite the survivor across its rung) or b_0 (trailing). From an opposite-rung position, a forward rail move allows only the next opposite-rung position or a trailing position. From a trailing position the zombie must follow. Both permit every forward survivor step. The departures in

$$b_1, b_2, \dots, b_{m-1}, b_0$$

therefore damage the entire second rail; the final listed departure is the move $b_0 \rightarrow b_1$. Each rail contributes exactly m departures, for $2m$ survivor moves in total. No return to the first rail is required. $\square$

Lemma 18. *None of $K_4, K_{3,3}, P_3, Q_3$ has full zombie damage.*

Proof. We give zombie strategies with damage upper bounds 0, 4, 2, 4, respectively. These bounds alone suffice; matching lower bounds are not needed here. Whenever a proposed survivor move permits capture on the following zombie turn, the zombie captures before the destination is damaged. Such deviations cannot enlarge the sets described below.

In K_4 any starting zombie vertex is universal, so capture precedes the first survivor move.

In $K_{3,3}$ label the parts $\{a_0, a_1, a_2\}$ and $\{b_0, b_1, b_2\}$. Start at a_0 . Unless capture occurs immediately, the survivor starts in $\{a_1, a_2\}$. Move to b_0 . Thereafter, as long as the survivor avoids imminent capture, it must alternate the two parts and avoid the occupied endpoint of the fixed edge $a_0 b_0$. The zombie alternates that edge as well. Each chosen zombie move is geodesic: before it the players are distinct vertices in one part, at distance two. All damaged vertices belong to $\{a_1, a_2, b_1, b_2\}$. A pass or other unsafe move is met by capture instead of continuing the alternation.

In P_3 , start at a_0 . A nonadjacent survivor start is b_1 or b_2 , and symmetry permits b_1 . Move the zombie to b_0 . The only survivor move that avoids imminent capture is the rung move to a_1 :

the remaining neighbor in the b -triangle is adjacent to b_0 . The zombie follows its own rung to a_0 , a legal distance-reducing move. Repeating this strategy confines safe departures to $\{a_1, b_1\}$.

For Q_3 , use binary coordinates and start the zombie at 000. We first describe a shadowing strategy from adjacent positions. If the zombie and survivor differ only in coordinate r , a safe survivor step changes a different coordinate; the zombie copies that change on its next move. That move is geodesic and preserves the difference in coordinate r . The survivor cannot change coordinate r without moving onto the zombie. A pass allows capture. Thus all subsequent damaged vertices are in the face with the survivor's coordinate r fixed, containing four vertices.

If the initial survivor distance is at most one, capture is immediate. At distance two, the first zombie move produces adjacent positions. The initial survivor vertex lies in the resulting fixed face and is credited on its first departure; shadowing keeps all later damaged vertices in that same face. At distance three, the survivor starts at 111. Let the zombie change coordinate one on every turn while the survivor keeps making safe edge moves. After each such zombie turn, the only safe neighbor available to the survivor is obtained by also changing coordinate one. Therefore these survivor positions alternate between 111 and 011, and both have coordinate three equal to one. If the survivor passes, on the next turn the zombie changes coordinate two instead. The players are now adjacent and differ only in coordinate three; adopt shadowing with $r = 3$. All earlier and later damaged vertices then lie in the same four-vertex face. If the survivor never passes, only the two alternating vertices are damaged. This proves the required upper bound in all cases. $\square$

Proof of Theorem 2 and Corollary 3. Each exceptional graph is excluded by Lemma 18. Every graph with a clean edge has full damage by Theorem 4, with bound (7). If no clean edge exists, the classification leaves only $K_4, K_{3,3}$, prisms, and Möbius ladders. After excluding P_3 and $P_4 = Q_3$, all remaining prisms are covered by Lemma 17. After excluding $M_6 = K_{3,3}$, all remaining Möbius ladders are covered by Lemma 16. Their bound of n moves is no larger than (7). The four exceptional orders are 4, 6, 6, 8, so every order at least ten satisfies the conjecture. $\square$

Remark 19 (Coverage and a known sensor obstruction). Let G_t be the ring formed from $t \geq 3$ disjoint diamonds $K_4 - e$ by joining consecutive degree-two ports cyclically. It is a bridgeless cubic graph on $4t$ vertices. Any cycle using an inter-diamond edge uses all t such edges and at least two internal edges per diamond, so these edges are clean. Theorem 4 therefore gives recurrent full coverage. Sardar et al. [10, Theorem 2] already proved that the metric dimension of G_t is t .

For completeness, the lower bound has a direct explanation: the two degree-three vertices within each diamond can be interchanged by an automorphism fixing every other vertex. A sensor set omitting that pair gives equal distances to its two vertices. Thus every set distinguishing all singleton locations must meet each of the t disjoint pairs. The same obstruction applies when a sensor reports the number of occupants at each distance: the two singleton populations have identical histograms, so the sensors do not distinguish all populations of mass at most h , for any integer $h \geq 1$. See [9] for that observation model. The sensor bound and ring construction are known; this remark only relates them to recurrent pursuit coverage. The companion retains the general involution proof, its ring specialization, and the resulting sensor-cardinality bound as supporting formalizations.

8 Verification, reproducibility, and formalization scope

The proof of Theorem 2 consists of the all-orders arguments above and the complete finite tables below. The tables can be inspected using Lemma 11. Executable checks are included to reduce transcription and case-checking risk. They do not turn a finite graph search into an infinite proof.

The core reduced-table verifier uses only the Python standard library. It reconstructs the five graphs from the stated definitions, checks every legal distance-decreasing zombie response, verifies the listed automorphisms, and checks coverage of every ordered port pair and every internal target. It accepts all 19 tasks and 121 rows and rejects 11 deliberately malformed variants, including a missing zombie branch, a nondecreasing rank, an unsafe move, and incorrect damage timing. A second retained certificate collection checks all 94 unreduced tasks and 472 rows.

Additional, nonessential regression checks test the explicit ladder strategies for parameters through 50 (24,280 tasks), the closed-family walks through order 200, 17 small whole-game instances, and 12 actual ambient gluings with loops or parallel edges in their quotients. A finite attachment audit checks the transition table through ladder parameter 12. The unbounded rows in Table 1 are justified by the distance argument in its proof, not by this cutoff. These structural audits use NetworkX; the core routing verifiers do not.

Third-party census audit. An author-supplied archive, generated by AI during review, contains separately written game and classification programs, census inputs, and execution logs. We preserved that archive and replayed its exact zombie-damage computations on all 4,681 connected cubic graphs through order 16. The only bridgeless graphs without full damage are the four stated exceptions; another 138 graphs without full damage have bridges. We also regenerated the connected cubic census using nauty’s `geng` [11] and replayed the structural audit on all 556,471 graphs through order 20. Its 1,608,255 nontrivial short-cycle components give zero classification violations; one-port components remain recorded as the unclassified one-port class. The supplied order-10-through-18 inputs agree byte for byte with the regenerated census.

At orders 18 and 20, the supplied complete scan uses a one-sided certificate for failure of full damage. For some initial zombie vertex, every survivor start has a target set of size at most two that cannot all be damaged. Every full-damage strategy would damage that set, so a positive certificate proves $\text{zdmg}(G) < n$; a zero filter result gives no conclusion. Our complete replay on all 41,301 and 510,489 graphs at those two orders reproduced 823 and 7,343 positive certificates, respectively; every flagged graph has a bridge. These scans do not verify full damage on the unflagged graphs. The supporting materials include the programs, supplied census data and numerical outputs, together with instructions and checksums for regenerating the complete census through order 20. The audit also supplies sampled composed-strategy trajectories; those trajectories are not universal adversarial certificates.

The earlier independent C++ check on 107 retained cubic graphs and 14 litmus or relabelling cases remains a separate regression. Independently adjoining literal triangles and squares gives 61 nonisomorphic cycle unions through order 24, without pruning by the proposed families; all belong to the classification. The ring regression checks 48 instances and their central-pair observation collisions. All of these computations have finite scope and supplement the all-orders written arguments.

Alignment with the original game. The general Lean game follows Davila’s Definition 1: the zombie starts first, the survivor observes that choice, the zombie takes the first turn, and every legal geodesic tie is adversarial. Passes are legal survivor moves. We omit an immediate-capture move onto the zombie: replacing it by a pass credits exactly the same source before forced capture, so this normalization does not weaken the survivor for the damage payoff, including the exception upper bounds. The completed survivor move credits its source; arrival alone is not damage, and later capture does not undo a completed departure. The full game imposes no distance-two restriction. For leaf-augmented routing components, a formal converse proves that any successful full-game exit

strategy from an adjacent position can be expressed in the local routing language. A separate Lean proof establishes the occupied-vertex normalization for every damage-only objective at the same move budget.

An independently written Lean executable and a separate C++ solver agree on zombie and ordinary-cop damage for 221 retained graphs. The inputs include all 143 connected graph-atlas graphs through order six, trees through order eight, paths and cycles through order ten, complete multipartite examples, subdivisions, and graphs with bridges or leaves. Every graph and distance table is also validated by ordinary Lean kernel reduction. Established numerical consequences of Davila’s Propositions 2–3 and 9, Theorems 4–6, 8, 10, 13 and 15, Corollaries 7, 11 and 16–18, and the one-pursuer portions of Proposition 19 were checked on the applicable inputs; Lemmas 12 and 14 received separate finite structural checks. No discrepancy was found. These executions are regression tests, not formal proofs of all of those original results. Multi-pursuer capture numbers and connected domination are outside this one-zombie implementation; no conjecture in the source is assumed.

End-to-end Lean formalization. The project uses Lean 4.32.0 [12] and Mathlib v4.32.0 [13]. The self-contained directory `palomar-full/` combines the complete main proof with the game library. The source theory for the zombie damage number, its game rules and Conjecture 24 is due to Randy Davila. Lennart Rudolph is the author of the formalization and the proof developed here, with the AI assistance disclosed below. Its main theorem `ZombieMain.full_damage_within_iff` quantifies over every finite simple graph and assumes only connectedness, degree three at every vertex, and absence of bridges. It proves the four-exception equivalence with the exact $2n(n+1)$ move bound. The separate declarations `full_damage_iff` and `davila_conjecture_24` give the unbounded formulation and the order-at-least-ten corollary.

The numerical interface defines `zombieDamageNumber` as the largest forceable distinct-vertex damage score. It proves the positive-threshold characterization, equivalence of bounded and eventual score forcing, and equivalence of value n with full damage for nonempty graphs. Thus `davila_conjecture_24_damage_number` gives the literal numerical conclusion $\text{zdmg}(G) = n$ from the original graph hypotheses.

The dependency chain proves the actual short-cycle classification for every ladder length, constructs faithful component models with identified external stubs, and instantiates every local strategy. It proves quotient routing, the $2n$ target-service bound, and initialization for every original zombie start. Closed-family relabellings and all four exception identifications are explicit graph isomorphisms. No decomposition, routing, initialization or full-damage premise is assumed by the main theorem.

The recurrent-coverage declaration constructs a fixed survivor controller. Every legal zombie reply preserves safety and the actual positions; on every infinite play, every vertex is a completed departure source beyond every finite time. Its progress proof rules out an infinite unfinished local route. Additional declarations prove the four-port and exterior-edge bounds, the expansion corollary, recurrent coverage of Möbius ladders, and the attributed diamond-ring sensor count.

The original `Std-only` game library also proves the complete local routing contracts, finite scheduling, bounded/eventual forcing equivalence, closed-family full damage, exception confinement, isomorphism invariance and observation obstructions. The registered package independently states 39 pursuit, structural and game-value results in `Challenge.lean`, selected by `comparator.json`. Twelve selected statements use transparent presentation wrappers that directly apply the original theorems. The three sensing results remain proved as supplementary library material. The separate `GameChallenge.lean` independently restates 24 game-library results. Neither

Challenge is imported by a proof. The solution modules contain no proof holes or project axioms. Finite proof obligations use ordinary kernel reduction. All 276 transitive axiom reports, including 13 proved presentation wrappers, use only `propext`, `Quot.sound`, and `Classical.choice`.

The arbitrary-game interface formulation explains the composition method; its complete cubic-graph application is proved directly in Lean. The illustrative ring’s structural link to coverage and the unselected historical results are supporting written arguments, rather than additional hypotheses in the formal main theorem. All 39 registered selections passed Comparator and NanoDa; the separate 24-statement comparison also passed locally. Formal statement fidelity, independent proof-checker results and exact source hashes are documented in `FORMALIZATION_STATUS.md` and the registry record linked below. Registry verification is distinct from independent human peer review.

Availability. Paper and supporting artifacts: doi:10.5281/zenodo.22667596. The full formalization is registered as Palomar PALOMAR-2026-09-10-000001, registry version 1. The immutable source snapshot is pinned to Git commit `f8f870fb1ce9`. The accompanying archives contain the $\text{\LaTeX}$ source, all policy data, reproducibility scripts, numerical results, checksums, the unified and component Lean projects, build and Comparator procedures, and citation metadata. The main verification commands, run from the extracted archive root, are

```
python3 verification/verify_reduced.py
python3 verification/reproduce.py
```

The reproduction command also checks fidelity between raw data and generated files; this is not Lean acceptance. The independent C++ and cycle-closure audits are selected with `-with-new-audits`. The ring checks are included in the default reproduction set. Full reproduction instructions and all-order/formal-scope boundaries are in `README.md` and `FORMALIZATION_STATUS.md`.

AI assistance and review status. The proof development, manuscript, certificate generation, and supporting software were prepared with substantial assistance from OpenAI ChatGPT. OpenAI Codex (Astra) assisted this revision with literature and citation searches, mathematical and source review, Lean proof development and build repairs, execution of verification tools, and manuscript and release preparation. Anthropic Fable 5.1 assisted in adversarial reviews. AI systems are not listed as authors. The named author is responsible for the submitted work.

The executions reported here are backed by retained inputs and logs. Additional AI-generated review comments supplied by the author informed the exposition; a subsequently supplied audit archive contains additional programs, inputs and logs. Its original reports and the checks replayed for this release are distinguished in the numerical-results documentation. The completed formalization verifies the main theorem and its graph-to-strategy dependencies. This mathematical preprint does not assert independent human verification or external peer review. The arguments and reproducibility materials are provided for independent scrutiny.

A Complete representative policies

The graph labels are those in Section 5. For each port p , the symbol p^* denotes its auxiliary leaf in the local model. In every task, p is the entry, E is the set of allowed exit ports, and t is the target or “none”. The initial state $(0, p^*, p)$ is bold. A row gives the current flag b , zombie vertex u , survivor vertex v , prescribed survivor destination w , and rank ρ .

For example, R1 begins at $(0, 2^*, 2)$ in $K_{2,3}$ and moves to 0. Its next zombie position is forced to 2. From $(1, 2, 0)$ the survivor moves to 3; the zombie may move to either 0 or 1. Both resulting rows exit at 3^* . Thus the table handles both zombie branches, not a selected play. The other rows are checked in exactly the same way using (2)–(6).

Table 3: The complete 19 representative policies (121 rows). Each initial row is bold.

b	u (zombie)	v (survivor)	w (move)	ρ
R1: $H = K_{2,3}$, $p = 2$, $E = \{3\}$, $t = \text{none}$				
0	2*	2	0	3
1	0	3	3*	1
1	1	3	3*	1
1	2	0	3	2
R2: $H = K_{2,3}$, $p = 2$, $E = \{3, 4\}$, $t = 0$				
0	2	0	3	2
0	2*	2	0	3
1	0	3	3*	1
1	1	3	3*	1
R3: $H = K_{3,3} - e$, $p = 0$, $E = \{3\}$, $t = \text{none}$				
0	0*	0	4	4
1	0	4	1	3
1	1	3	3*	1
1	2	3	3*	1
1	4	1	3	2
1	5	1	3	2
R4: $H = K_{3,3} - e$, $p = 0$, $E = \{3\}$, $t = 1$				
0	0	4	1	3
0	4	1	3	2
0	5	1	3	2
0	0*	0	4	4
1	1	3	3*	1
1	2	3	3*	1
R5: $H = K_{3,3} - e$, $p = 0$, $E = \{3\}$, $t = 4$				
0	0	4	1	3
0	0*	0	4	4
1	1	3	3*	1
1	2	3	3*	1
1	4	1	3	2
1	5	1	3	2

Continued on the next page

Representative policies, continued

b	u (zombie)	v (survivor)	w (move)	ρ
R6: $H = P_3 - e_\Delta$, $p = 0$, $E = \{1\}$, $t = \text{none}$				
0	0*	0	2	3
1	0	2	1	2
1	2	1	1*	1
R7: $H = P_3 - e_\Delta$, $p = 0$, $E = \{1\}$, $t = 2$				
0	0	2	1	2
0	0*	0	2	3
1	2	1	1*	1
R8: $H = P_3 - e_\Delta$, $p = 0$, $E = \{1\}$, $t = 3$				
0	0	3	4	3
0	0*	0	3	4
1	3	4	1	2
1	4	1	1*	1
R9: $H = P_3 - e_\Delta$, $p = 0$, $E = \{1\}$, $t = 4$				
0	0	3	4	3
0	3	4	1	2
0	0*	0	3	4
1	4	1	1*	1
R10: $H = P_3 - e_\Delta$, $p = 0$, $E = \{1\}$, $t = 5$				
0	0	2	5	7
0	2	5	4	6
0	3	5	2	3
0	0*	0	2	8
1	0	2	1	2
1	1	4	3	5
1	2	1	1*	1
1	3	0	2	3
1	4	1	1*	1
1	4	3	0	4
1	5	2	1	2
1	5	4	1	2
R11: $H = Q_3 - v$, $p = 3$, $E = \{5\}$, $t = \text{none}$				
0	3*	3	1	3
1	1	5	5*	1
1	3	1	5	2

Continued on the next page

Representative policies, continued

b	u (zombie)	v (survivor)	w (move)	ρ
R12: $H = Q_3 - v$, $p = 3$, $E = \{5, 6\}$, $t = 0$				
0	1	0	2	3
0	2	0	1	3
0	3	1	0	4
0	3*	3	1	5
1	0	1	5	2
1	0	2	6	2
1	1	5	5*	1
1	2	6	6*	1
1	3	1	5	2
1	3	2	6	2
1	4	5	5*	1
1	4	6	6*	1
R13: $H = Q_3 - v$, $p = 3$, $E = \{5, 6\}$, $t = 1$				
0	3	1	5	2
0	3*	3	1	3
1	1	5	5*	1
R14: $H = Q_3 - v$, $p = 3$, $E = \{5, 6\}$, $t = 4$				
0	0	4	5	2
0	1	0	4	3
0	2	0	4	3
0	3	1	0	4
0	5	4	6	2
0	6	4	5	2
0	3*	3	1	5
1	1	5	5*	1
1	4	5	5*	1
1	4	6	6*	1
R15: $H = Q_3 - e$, $p = 0$, $E = \{1\}$, $t = \text{none}$				
0	0*	0	2	4
1	0	2	3	3
1	2	3	1	2
1	3	1	1*	1
R16: $H = Q_3 - e$, $p = 0$, $E = \{1\}$, $t = 2$				
0	0	2	3	3
0	0*	0	2	4
1	2	3	1	2
1	3	1	1*	1
R17: $H = Q_3 - e$, $p = 0$, $E = \{1\}$, $t = 3$				
0	0	2	3	3
0	2	3	1	2
0	0*	0	2	4
1	3	1	1*	1

Continued on the next page

Representative policies, continued

b	u (zombie)	v (survivor)	w (move)	ρ
R18: $H = Q_3 - e$, $p = 0$, $E = \{1\}$, $t = 6$				
0	0	2	6	5
0	2	6	4	4
0	4	6	2	4
0	0*	0	2	6
1	0	2	3	3
1	0	4	5	3
1	2	3	1	2
1	3	1	1*	1
1	4	5	1	2
1	5	1	1*	1
1	6	2	3	3
1	6	4	5	3
1	7	3	1	2
1	7	5	1	2
R19: $H = Q_3 - e$, $p = 0$, $E = \{1\}$, $t = 7$				
0	0	2	3	9
0	2	3	7	8
0	3	7	5	7
0	6	7	3	3
0	0*	0	2	10
1	0	2	3	3
1	1	5	4	6
1	2	3	1	2
1	3	1	1*	1
1	4	0	2	4
1	5	1	1*	1
1	5	4	0	5
1	6	2	3	3
1	7	3	1	2
1	7	5	1	2

References

- [1] R. Davila. *The Zombie Damage Number of a Graph*. arXiv:2607.16382v1 [math.CO], 17 July 2026. doi:10.48550/arXiv.2607.16382.
- [2] D. Cox and A. Sanaei. The damage number of a graph. *Australasian Journal of Combinatorics* **75**(1), 1–16, 2019. https://ajc.maths.uq.edu.au/pdf/75/ajc_v75_p001.pdf.
- [3] M. A. Huggan, M.-E. Messinger, and A. Porter. The damage number of the Cartesian product of graphs. *Australasian Journal of Combinatorics* **88**(3), 362–384, 2024. https://ajc.maths.uq.edu.au/pdf/88/ajc_v88_p362.pdf.
- [4] M. A. Huggan, M.-E. Messinger, and A. Porter. Corrigendum to: The damage number of the Cartesian product of graphs. *Australasian Journal of Combinatorics* **91**(1), 217–218, 2025. https://ajc.maths.uq.edu.au/pdf/91/ajc_v91_p217.pdf.
- [5] M. Stojaković and L. Wulf. *On the multi-robber damage number*. arXiv:2209.10965v4 [math.CO], 2022; revised 22 June 2026. doi:10.48550/arXiv.2209.10965.
- [6] V. Gledel, W. B. Kinnersley, B. Patkós, and M. Stojaković. *On the damage number of graphs*. arXiv:2608.15226v1 [math.CO], 15 August 2026. doi:10.48550/arXiv.2608.15226.
- [7] F. Liu and A. Miller. Replacing Cops with Zombies. In *13th International Conference on Fun with Algorithms (FUN 2026)*, *Leibniz International Proceedings in Informatics* **366**, 30:1–30:12, 2026. doi:10.4230/LIPIcs.FUN.2026.30.
- [8] D. Cushing, R. Kangaslampi, V. Lipiäinen, S. Liu, and G. W. Stagg. *The Graph Curvature Calculator and the curvatures of cubic graphs*. *Experimental Mathematics* **31**(2), 583–595, 2022 (published online 2019). doi:10.1080/10586458.2019.1660740. Version-specific locators refer to arXiv:1712.03033v2, 22 December 2017.
- [9] L. Rudolph. *Distance-Shell Tomography on Graphs: Integer Trades and Optimal Grid Sensing*. Zenodo preprint, version 1.0.0, 2026. doi:10.5281/zenodo.22404456.
- [10] M. S. Sardar, S.-J. Xu, M. Cancan, M. R. Farahani, M. Alaeiyan, and S. V. Patil. Computing Metric Dimension of Two Types of Claw-free Cubic Graphs with Applications. *Journal of Combinatorial Mathematics and Combinatorial Computing* **119**, 163–174, 2024. doi:10.61091/jcmcc119-17.
- [11] B. D. McKay and A. Piperno. Practical graph isomorphism, II. *Journal of Symbolic Computation* **60**, 94–112, 2014. doi:10.1016/j.jsc.2013.09.003.
- [12] L. de Moura and S. Ullrich. The Lean 4 Theorem Prover and Programming Language. In *Automated Deduction—CADE 28*, 625–635, 2021. doi:10.1007/978-3-030-79876-5_37.
- [13] The mathlib Community. The Lean Mathematical Library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, 367–381, 2020. doi:10.1145/3372885.3373824.