SURVIVABLE DIRECTED SUPPORTS FOR SYNCHRONOUS LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We study minimum-cost directed communication supports that retain prescribed coordinator–worker connections after bounded arc failures, subject to a limit of ℓ provisioned arcs. For requirements of multiplicities $b_1, \ldots, b_q$, an integral-flow dynamic program stores only boundary imbalances and the number of selected arcs. On a supplied width-τ decomposition, its running time is $2^{O((\tau+1)\sum_g \log(2b_g+1))}\,\mathrm{poly}(N, \ell, L)$. The bounded state is complete because each requirement admits a witness consisting of simple arc-disjoint paths, even though the algorithm permits circulations. Rooted localization then gives a $2^{O(\ell^2)}\,\mathrm{poly}(N, L)$ algorithm on planar and fixed apex-minor-free underlays. On a 40-instance benchmark, the flow DP completes 22 instances, the labelled-path DP completes 20, and MILP completes all 40 under common limits. A separate provisioning study shows how shared resilient supports preserve the prescribed synchronous learning updates under an explicit arc-failure model.

1 INTRODUCTION

A synchronous learning round depends on a concrete communication condition. Every designated worker must receive the current model and return its update before aggregation. Losing either connection can stop the round or change the training population. We study the offline decision that precedes this computation. Given a directed underlay, nonnegative provisioning costs, a shared arc budget, and a failure tolerance for each worker, choose a minimum-cost support that retains both required directions. The decision concerns which resources are provisioned. The resulting support does not prescribe a new learning rule.

Topology is a meaningful design variable in distributed learning. Marfoq et al. (2020) optimize throughput on directed underlays, while You & Pu (2024) use directed trees to distribute parameter and gradient information. Here the objective is worst-case reachability after arc deletions. These objectives differ, and neither cited result implies that a learning deployment has bounded local treewidth. Our contribution is an algorithmic result under an explicit structural assumption, with synchronous learning providing the communication semantics.

For a root r and worker t, requiring b_t arc-disjoint directed paths is equivalent to preserving the connection after any $b_t - 1$ arc deletions. Even uniform outward demand two gives W[1]-hardness in the number of selected arcs on unrestricted digraphs (Feldmann et al., 2025, Theorem 1.6). We restrict attention to the width of the root's relevant neighborhood. An inclusion-minimal feasible support with at most ℓ arcs lies in the radius-ℓ ball of the underlying undirected graph. Its directional path demand is at most ℓ, and the joint demand is at most 2ℓ. A decomposition of that ball permits optimization over integral flow certificates.

The main algorithm uses one binary flow per directional requirement. Capacities apply separately to each flow, while provisioning cost is paid on their union. At a separator, only each flow's imbalance matters. Individual path identities and the pairing of their boundary endpoints can be discarded. The difficulty is bounding those imbalances without losing an optimum. A witness consisting of b_g simple arc-disjoint paths has indegree and outdegree at most b_g at every vertex. Its restriction to any processed arc set therefore has imbalance in $[-b_g, b_g]$. This observation gives a finite state space and removes the logarithmic width factor incurred by ordered path pairings.[1]

[1]We thank a referee for suggesting the independent-flow state used here.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Contribution and evidence. The algorithmic result is an explicit bound in the width and the demand profile for a shared, directed, nonuniform design problem. Its state count distinguishes the number of requirements from their multiplicities. Corollary 8 gives the conditional execution consequence for a synchronous protocol. It is an application of surviving reachability, not a new convergence theorem. The experiments address two separate questions. Matched comparisons with a labelled-path DP and MILP identify where the smaller representation changes practical tractability. A distinct MILP study evaluates the provisioning objective and its participation consequences. It supplies no evidence that the DP scales to the larger design instances.

Relationship to existing theorems. Feldmann et al. (2025, Theorems 1.4–1.5) establish qualitative tractability for undirected vertex-connectivity survivable network design (VC-SNDP), with arbitrary terminal-pair demands, on locally bounded-treewidth graphs and for treewidth plus total demand. Their Theorem 1.6 gives the rooted directed hardness result just cited. Jones et al. (2017) instead consider rooted directed reachability with unit demand and parameterize by selected nonterminals. Appendix F compares the models and constructive guarantees theorem by theorem. Neither undirected vertex connectivity nor the nonterminal parameter is interchangeable with our model. Nevertheless, localization and a logical optimization theorem already provide a route to qualitative fixed-parameter tractability (Arnborg et al., 1991). We do not claim that general route as new. The result here makes the directed shared-cost state space and exponent explicit. Chalermsook et al. (2018, Theorem 4 and Lemma 8) also give exact rooted survivable-design algorithms and a connection-profile state bound in width and maximum demand. Our bound instead exposes the dependence on each requirement multiplicity. These parameterizations are not ordered, so this is not a claim of uniform improvement over prior bounded-width algorithms. Directed approximation algorithms address a different objective regime (Grandoni & Laekhanukit, 2017; Chan et al., 2020).

2 Model and algorithmic guarantee

Let $D = (V, A)$ be a finite directed multigraph, $r \in V$ a coordinator, and $T \subseteq V \setminus \{r\}$ a worker set. Each arc has a nonnegative rational provisioning cost c_e. Parallel and antiparallel arcs are distinct resources. Self-loops can be deleted. Write $N = |V| + |A|$, let L denote the binary encoding length of the numerical input, and let $U(D)$ be the simple underlying undirected graph. For $F \subseteq A$, write $D[F] = (V, F)$ and $c(F) = \sum_{e \in F} c_e$. Provisioning cost is paid once per selected arc and is distinct from per-message communication cost.

Definition 1 (Survivable provisioning)**.** For an integer budget $\ell \geq 0$, outward requirements $b_t^+ \geq 1$, and optional inward requirements $b_t^- \geq 1$, find

$$\min_{F \subseteq A} c(F) \quad \text{subject to} \quad |F| \leq \ell, \qquad \lambda_F(r,t) \geq b_t^+, \qquad \lambda_F(t,r) \geq b_t^- \quad (t \in T), \tag{1}$$

where $\lambda_F(u, v)$ is the maximum number of pairwise arc-disjoint directed u–v paths. The outward-only problem omits the last constraints. An infeasible instance must be reported as such.

Different requirements do not impose simultaneous flow capacities on each other. Their path certificates describe survivability, and messages may be scheduled sequentially. An arc is one directed resource in the failure model. The guarantee counts deleted arcs, not physical incidents. Two logical channels carried by the same cable do not constitute separate protection against failure of that cable. Such an incident deletes every affected arc and consumes that many units of the failure budget. Antiparallel arcs are likewise separate resources only under this arc-level model. Shared failure domains and node failures require a different specification.

Let $B_\ell(r) = \{v \in V : \mathrm{dist}_{U(D)}(r, v) \leq \ell\}$. Define $p^+ = \sum_t b_t^+$, $p^- = \sum_t b_t^-$, and $p = p^+ + p^-$, taking $p^- = 0$ when inward requirements are absent. Index the q ordered requirements by g, write their multiplicities as b_g, and set $\kappa = \sum_{g=1}^{q} \log_2(2b_g + 1)$. Thus $q \leq p$ and $\kappa \leq p \log_2 3$. Empty requirements have the empty optimum. We may replace ℓ by $\min\{\ell, |A|\}$.

Theorem 2 (Instance-wise optimization)**.** *Given a width-τ tree decomposition of $U(D)[B_\ell(r)]$ with polynomially many bags, problem* (1) *can be solved exactly in*

$$2^{O((\tau+1)\kappa)} \, \mathrm{poly}(N, \ell, L) \tag{2}$$

time, including reconstruction of an optimal support. Since $\kappa \le p \log_2 3$, this is also $2^{O(p(\tau+1))}\,\mathrm{poly}(N, \ell, L)$. The polynomial also accounts for reading and normalizing the supplied decomposition. Every feasible instance satisfies $p^+ \le \ell$ and $p^- \le \ell$, hence $p \le 2\ell$. In the outward-only case, $p \le \ell$.

A class $\mathcal{C}$ has a *constructive local-treewidth bound* Λ if a width-$\Lambda(\rho)$ decomposition of every radius-ρ ball in a graph of the class can be found in $f_{\mathcal{C}}(\rho)N^{O(1)}$ time. This is an algorithmic class hypothesis. It is not an additional numerical input parameter and does not replace the width actually supplied to Theorem 2.

Corollary 3 (Uniform class guarantee)**.** *On such a class, both provisioning problems admit an exact algorithm in*

$$f_{\mathcal{C}}(\ell)N^{O(1)} + 2^{O(\ell(\Lambda(\ell)+1))}\,\mathrm{poly}(N, \ell, L).$$

For planar and every fixed apex-minor-free underlying class, a constructive linear local-treewidth bound gives $2^{O(\ell^2)}\,\mathrm{poly}(N, L)$ time.

For this corollary, combine the structural width bound of Demaine & Hajiaghayi (2004) with the $2^{O(k)}N$-time constant-factor decomposition algorithm of Bodlaender et al. (2016), taking $k = O(\ell)$. Thus the decomposition cost fits within the stated planar and apex-minor-free running time.

The distinction between local and global width is substantive. A large planar grid has large global treewidth, but its radius-ℓ ball has width $O(\ell)$. Conversely, a small ball can still contain many vertices, as in a star, so the localization does not establish a kernel with few vertices. An arbitrary sparse or geometric deployment need not meet the hypothesis. A supplied decomposition makes the instance-wise condition verifiable without asserting that every possible deployment belongs to a favorable graph class.

3 Why a small support is local

Lemma 4 (Demand accounting)**.** *Every feasible support F satisfies $p^+ \le |F|$ and $p^- \le |F|$.*

Proof. For each outward requirement, choose b_t^+ arc-disjoint r–t paths. Their final arcs are distinct for fixed t, and final arcs belonging to distinct terminals have distinct heads. Thus F contains at least $\sum_t b_t^+$ arcs. For inward requirements, use the first arcs of the t–r paths. These are distinct within a requirement and have distinct tails across terminals. The two counts may overlap, which is why their sum is bounded by $2|F|$, not by $|F|$. □

Lemma 5 (Localization)**.** *Every arc of an inclusion-minimal feasible support F with $|F| \le \ell$ has both endpoints in $B_\ell(r)$.*

Proof. Choose, for every requirement, the prescribed number of simple arc-disjoint witness paths in $D[F]$. Their union H is feasible and is contained in F. Inclusion-minimality forces $H = F$. Every selected arc therefore lies on a simple witness path, which either starts or ends at r and uses at most $|F| \le \ell$ arcs. Each vertex on that path has undirected distance at most ℓ from r. □

Nonnegative costs ensure that an optimal support has an equally good inclusion-minimal feasible subset. Lemma 5 therefore preserves the optimum when we restrict to $D[B_\ell(r)]$. Any positive-demand terminal outside this ball causes immediate rejection. Lemma 4 is checked before allocating flow coordinates, using sums saturated at $\ell + 1$ so that binary-encoded large demands do not cause an artificial blowup.

For each requirement $g = (s_g, t_g, b_g)$, choose a binary arc vector $y_g \in \{0,1\}^A$ with divergence

$$\beta_g(v) = b_g\big(\mathbf{1}\{v = s_g\} - \mathbf{1}\{v = t_g\}\big).$$

Here divergence is outgoing flow minus incoming flow. Integral flow decomposition gives b_g arc-disjoint s_g–t_g paths after deleting circulations. Conversely, the union of such paths gives a binary flow. Requirements have separate unit capacities, so they can use the same original arc. The purchased support is $F = \bigcup_g \mathrm{supp}(y_g)$, with $|F| \le \ell$. This is equivalent to the provisioning problem after taking an inclusion-minimal optimum.

4 DYNAMIC PROGRAMMING WITH BOUNDARY FLOWS

The construction works for any list of ordered requirements, whether or not they share a root. Rootedness is used only for the localization and demand bounds in Section 3.

4.1 DECOMPOSITION AND THE BOUNDED-WITNESS ARGUMENT

A tree decomposition has bags $B_x \subseteq V$ on a tree, with every vertex present, every original arc's endpoints together in a bag, and the bags containing any fixed vertex connected. Its width is $\max_x |B_x| - 1$. Root and normalize the decomposition to have empty leaves and root, introduce-vertex nodes, forget-vertex nodes, equal-bag two-child joins, and introduce-arc nodes. Each original arc is assigned to exactly one introduce-arc node containing its endpoints. Parallel and antiparallel copies remain separate. The refinement preserves width and has polynomial size.

Let V_x be the vertices appearing in the subtree at x and A_x the arcs introduced there, including at x. For a partial binary flow $y_g \in \{0,1\}^{A_x}$, define

$$d_{g,x}(v) = \sum_{e \in A_x \cap \delta^+(v)} y_{g,e} - \sum_{e \in A_x \cap \delta^-(v)} y_{g,e}. \tag{3}$$

Every vertex in $V_x \setminus B_x$ already has its final divergence $\beta_g(v)$. Bag vertices may have unfinished balances.

Lemma 6 (Bounded witness)**.** *There is an optimal support with binary witnesses for which $|d_{g,x}(v)| \le b_g$ at every decomposition node x and every bag vertex v.*

Proof. Choose an inclusion-minimal optimal support and, for each requirement, b_g simple arc-disjoint witness paths. Their union over all requirements equals the support by minimality. For a fixed requirement, each vertex has at most b_g incoming and at most b_g outgoing witness arcs, since a simple path visits it at most once. Restricting to any subset A_x leaves both counts between zero and b_g. Their difference lies in $[-b_g, b_g]$. This holds simultaneously for every node and every requirement. □

The lemma bounds a complete witness under every restriction. It does not assert that every feasible partial flow has bounded imbalance. Some circulation-rich assignments will be discarded, and the table invariant below makes that restriction explicit.

4.2 STATE AND TRANSITIONS

A state at x is

$$(a, d), \qquad 0 \le a \le \ell, \qquad d \in \prod_{g=1}^{q} \{-b_g, \dots, b_g\}^{B_x}. \tag{4}$$

Its value $\mathrm{DP}_x(a, d)$ is the minimum union cost among partial binary flows on A_x with exactly a used arcs, boundary divergence d, and final divergence β_g at forgotten vertices. We admit only histories whose restriction to every descendant bag also respects that bag's imbalance bounds. An unrealized entry is $+\infty$. No path components, pairings, or degree bounds are stored.

Leaf and introduce vertex. The empty leaf has value zero at $(0, \varnothing)$. When introducing v, append a zero imbalance for v in every requirement. No incident arc occurs in the child subtree by the connectedness condition on bags. Cost and cardinality are unchanged.

Introduce arc. For $e = (u, v)$, choose the subset $J \subseteq \{1, \dots, q\}$ of requirements that use the arc. For each $g \in J$, increase $d_g(u)$ by one and decrease $d_g(v)$ by one. Reject a coordinate outside $[-b_g, b_g]$. If $J \neq \varnothing$, increase a by one and add c_e once. If $J = \varnothing$, change nothing. Reject $a > \ell$ and minimize over predecessors. One binary choice per requirement enforces its unit arc capacity. There is no constraint on $|J|$.

Forget vertex. When forgetting v, require $d_g(v) = \beta_g(v)$ for every g, then delete its coordinates. All incident arcs have already been introduced, so these are the final source, sink, or conservation equations. In particular, a source cannot be forgotten with zero imbalance when its demand is positive.

Join. For children y, z with common bag B_x, add imbalances and purchased-arc counts. Discard an out-of-range coordinate or a count above ℓ, and set

$$\mathrm{DP}_x(a, d) = \min_{\substack{a_y + a_z = a \\ d_y + d_z = d}} \big(\mathrm{DP}_y(a_y, d_y) + \mathrm{DP}_z(a_z, d_z)\big). \tag{5}$$

The introduced arc sets of the children are disjoint. Thus capacities remain binary, while cost and cardinality add without a correction term. The same graph vertex may occur in both children only if it belongs to their common bag.

A worked join. For one unit requirement, let the bag be $\{a, b, c\}$. Suppose the left child's segments are $s \to a$ and $b \to c$, with s already forgotten, and the right child's segments are $a \to b$ and $c \to t$, with t forgotten. In bag order, the left imbalance is $(-1, 1, -1)$ and the right imbalance is $(1, -1, 1)$. Their sum is zero, so all three bag vertices satisfy conservation. The union carries a unit s–t flow. No information about how the segments are paired is needed. If two child segments instead close a directed cycle, the join may retain it. A cycle has zero divergence and can be removed from a completed flow without destroying its required value.

At the empty root, minimize over all states $(a, \varnothing)$ with $a \le \ell$. Predecessor pointers recover the support.

4.3 Correctness and complexity

Lemma 7 (Table invariant)**.** *Every entry equals the minimum over the admissible histories specified above.*

Proof. Induct over the normalized decomposition. The leaf is the unique empty assignment. At an introduce-vertex node the new coordinates are necessarily zero. An introduce-arc assignment is determined by its set of users J, and its divergence and union charge change exactly as stated. Removing that arc recovers its admissible child history. Forgetting checks precisely the newly finalized balance equations and changes no arc. At a join, restricting a parent history to the two disjoint child arc sets gives the child histories. Conversely, any compatible pair of child histories forms a parent assignment whose divergences, cost, and cardinality add. Forgotten vertices lie in only one child subtree, so their equations remain valid. Each transition also checks the new bag's bounds. It therefore preserves exactly the restriction imposed on all descendant histories. Finally, future transitions depend only on the stored divergence and counter, so retaining a minimum-cost realization of each state is safe. □

For soundness, every accepted root assignment satisfies the full divergence equations and uses each original arc at most once per requirement. Integral flow decomposition yields b_g arc-disjoint paths for requirement g, possibly together with circulations. Deleting circulations from its certificate only removes arc uses. It cannot increase the union cost or cardinality or harm another requirement's certificate. Hence the recovered support is feasible. For completeness, Lemma 6 supplies an optimal path witness admitted at every node, so the root minimum is no larger than the optimum. Soundness gives the reverse inequality.

For $|B_x| \le \tau + 1$, the number of states is at most

$$S = (\ell + 1) \prod_{g=1}^{q} (2b_g + 1)^{\tau+1} = (\ell + 1) 2^{(\tau+1)\kappa}. \tag{6}$$

An arc node considers 2^q user sets per child state, and a naive join considers at most S^2 child pairs. Both fit within $2^{O((\tau+1)\kappa)}\,\mathrm{poly}(N, \ell, L)$, since $q \le \kappa$ and the decomposition has polynomial size. Nonnegative rational costs can be added and compared with polynomial bit complexity. This proves Theorem 2. Lemma 4 and the constructive local-width bound then prove Corollary 3.

For uniform multiplicity b, $\kappa = q \log_2(2b+1)$. The algorithm is thus singly exponential in width times the number of requirements when b is fixed, and polynomial in b for fixed width and requirement count. Total demand $p = qb$ alone hides this distinction. The flow representation improves the labelled-path upper bound $2^{O(p(\tau+1)\log(\tau+2))}$, but no lower bound establishes that the new state is minimal. The reduction also relies on independent requirement capacities. Shared traffic capacities, prescribed path lengths, and disjointness between different workers are different constraints.

5 WHAT THE DESIGN GUARANTEES FOR LEARNING

Consider a synchronous coordinator algorithm with prescribed worker weights $w_t \geq 0$, $\sum_t w_t = 1$. At round k, worker t receives θ_k, computes an update $u_t(\theta_k, \xi_{t,k})$, and returns it. The coordinator performs a fixed aggregation

$$\theta_{k+1} = \mathcal{A}_k\big(\theta_k, (u_t(\theta_k, \xi_{t,k}))_{t\in T}\big). \tag{7}$$

This includes full-participation gradient descent and fixed-participation model averaging (McMahan et al., 2017). The root and workers themselves do not fail. During each round a set of at most f arcs is unavailable and remains fixed until the round finishes. Assume the surviving routes can be found, messages on them are delivered exactly, duplicate messages are suppressed, and the coordinator waits for all prescribed updates. There is no deadline or requirement that simultaneous messages fit unit capacities.

Corollary 8 (Conditional execution preservation)**.** *If F satisfies* (1) *with $b_t^+, b_t^- \geq f + 1$ for every worker, every round can complete under these assumptions. For the same initialization and worker randomness, its iterates equal those of* (7) *on fault-free communication. This holds even when each round's failure set depends on the previous history.*

Proof. For a fixed round, at most f deletions leave one path from r to each t and one from t to r. Sending messages sequentially along simple surviving paths gives a finite communication schedule. If both executions start the round at θ_k, each worker receives the same input and uses the same random variable, hence returns the same update. Aggregating the same messages in the same order gives the same next iterate. Induction proves the claim, regardless of how the next failure set is selected. □

This corollary identifies the communication precondition preserved by the support. It preserves the intended training population and update rule. Any convergence statement for the reference iterates transfers in completed-round count, without a new convergence-rate claim. Sequential routing uses at most $2|T|\ell$ arc transmissions per round for one outward and one inward message per worker, excluding failure discovery and protocol overhead. If each such transmission takes at most Δ, this schedule takes at most $2|T|\ell\Delta$ communication time. It is an existence bound, not an optimized throughput guarantee.

For comparison, if unavailable workers are skipped, a gradient method may instead use

$$\widetilde{g}_k = \frac{\sum_{t\in R_k} w_t g_t(\theta_k)}{\sum_{t\in R_k} w_t}, \qquad R_k = \{t : \lambda_{F-S_k}(r,t), \lambda_{F-S_k}(t,r) \geq 1\}.$$

This differs from $\sum_t w_t g_t(\theta_k)$ when local gradients differ. The distinction is particularly transparent under heterogeneous data, but no independence of failures and data is assumed by Corollary 8. It does not establish fault tolerance for the decentralized dynamics of You & Pu (2024), secure aggregation, Byzantine workers, or arbitrary packet loss during a round.

6 ALGORITHM PERFORMANCE AND PROVISIONING CONSEQUENCES

The studies separate the proposed DP from the design objective. The first compares algorithms on identical inputs. The remaining studies use the independent flow formulation in Appendix C to examine shared provisioning and participation. Their conclusions concern the model, not DP scalability.

6.1 FLOW STATES, PATH PAIRINGS, AND MILP

Both DPs agree with exhaustive enumeration and the independent flow formulation on 88 small cases, including zero costs, cycles, parallel arcs, infeasibility, sharing, and opposite directions. Seventeen cases contain joins. The flow DP additionally agrees with exhaustive enumeration on 742 instance–budget combinations from 100 further multigraphs. The tests exercise all five node types, including two-child joins.

The performance comparison uses 40 grid instances, crossing four sizes, two cost seeds, and five requirement patterns. The budget is minimum feasible cardinality plus four. Every solver has a ten-second limit, and each DP table has a 50,000-state cap. Both DPs use the same deterministic decomposition procedure. Times include decomposition construction and exclude imports.

Grid	τ	Flow	Pairing	MILP	Flow ms	Pairing ms	MILP ms
2×4	2	6/10	6/10	10/10	0.7	1.1	3.3
2×10	2	6/10	6/10	10/10	2.1	7.1	4.5
3×4	3	6/10	4/10	10/10	1.6	5.8	4.0
4×5	4	4/10	4/10	10/10	11.1	247.4	4.5

Table 1: Matched 40-instance comparison. Completion counts include all instances. Time medians use the identical subset completed by all three solvers within each grid family. A ten-second limit applies to every solver and a 50,000-state cap to each DP table.

The flow DP completes 22 of 40 cases, compared with 20 for pairings and 40 for MILP. Every completed optimum agrees. The two additional completions have width three and two requirements of multiplicity two. The new state is also substantially faster on some shared completions, including the width-four single requirement of multiplicity two. Neither DP completes the six-requirement cases. Thus the improved exponent has a measurable implementation benefit, but does not establish general superiority over MILP.

To separate requirement count from multiplicity, we specify 36 additional cases before running the comparison. Two grid widths, three requirement counts $q \in \{1, 2, 6\}$, three uniform multiplicities $b \in \{1, 2, 3\}$, and two budget slacks are crossed. Two parallel copies per directed arc permit the larger demands without changing underlying width.

q	b	p	κ	Flow	Pairing	MILP
1	1	1	1.58	4/4	4/4	4/4
1	2	2	2.32	4/4	4/4	4/4
1	3	3	2.81	4/4	4/4	4/4
2	1	2	3.17	4/4	4/4	4/4
2	2	4	4.64	4/4	2/4	4/4
2	3	6	5.61	2/4	0/4	4/4
6	1	6	9.51	0/4	1/4	4/4
6	2	12	13.93	0/4	0/4	4/4
6	3	18	16.84	0/4	0/4	4/4

Table 2: Prespecified stress cases, with two widths and two budget slacks for each demand profile. Each arc has two parallel copies. Entries count completed optima out of all four settings. Full parameters and stopping times appear in Appendix G.

The flow and pairing DPs complete 22 and 19 of these cases, respectively, while MILP completes all 36. Four cases are completed only by the flow DP and one only by pairings. The latter has width two, six unit requirements, and the minimum feasible budget. Flow states permit circulation-containing assignments that the pairing DP excludes, so a better worst-case state bound need not give fewer realized states on every input. Appendix G reports the full parameter map and all failures, together with table sizes, transition counts, join work, time, and resident memory.

6.2 Shared provisioning under a common budget

We retain 30 bidirected grids of sizes 3×4, 4×5, and 5×6, with ten cost seeds each, costs in $\{1, \ldots, 10\}$, a corner coordinator, and the other corners as workers. Every worker requires two paths per direction. Budgets are 24, 32, and 40. All joint supports are computed by SciPy/HiGHS with zero reported relative gap and meet the prescribed connectivity demands. Independently optimal requirement unions and separately optimized direction unions exceed the common budget on every instance, so their costs are reported only as unbudgeted comparisons in Appendix D.

A descending-cost reverse-delete heuristic prunes an independent resilient union while preserving every requirement. It meets the common budget on 10/10, 3/10, and 1/10 trials. On those same

feasible trials, its mean costs are 127.2, 168.0, and 208.0, compared with 121.8, 157.0, and 185.0 for joint optimization. These are matched comparisons, not ratios averaged over different feasible subsets. The heterogeneous feasibility rates remain part of the result.

Failure percentages depend strongly on the sampling set. For the nominal joint supports, mean full-round retention is 66.5%, 73.5%, and 78.6% when each underlay arc is deleted in turn. It is zero at every size when conditioning on hitting a selected arc. The resilient supports retain every worker under both conventions. Table 10 reports both distributions for both nominal constructions. Selected-support conditioning is a vulnerability diagnostic whose sample space depends on the design, not a shared deployment failure law.

6.3 Participation under random failures

On the 3×4 instance with cost seed zero, we compare budget-feasible joint resilient, pruned resilient, nominal joint, and nominal terminalwise supports. Their respective arc counts are 20, 24, 10, and 15, and their costs are 119, 144, 59, and 80. A ten-class linear softmax model uses the 1,797-example digits dataset (Alpaydin & Kaynak, 1998), ten stratified train/test splits, and 200 attempted rounds. Workers receive either approximately balanced class proportions or class-wise Dirichlet proportions with concentration 0.5. Each round deletes one uniformly sampled underlay arc, independently of data and performance; the same failure sequence is used for every support. Unavailable workers are skipped and survivor weights renormalized. If none are available, the model is unchanged.

Support	Balanced accuracy	Heterogeneous accuracy	Full (%)	Workers (%)
Healthy	94.56 ± 0.62	94.56 ± 0.62	100.0	100.0
Joint resilient	94.56 ± 0.62	94.56 ± 0.62	100.0	100.0
Pruned resilient	94.56 ± 0.62	94.56 ± 0.62	100.0	100.0
Nominal joint	94.04 ± 0.82	94.04 ± 0.82	69.8	69.8
Nominal terminalwise	94.51 ± 0.68	94.29 ± 0.83	58.8	81.3

Table 3: Uniform per-round underlay-arc failures on the fixed 3×4 instance. Every support meets the common budget 24. Accuracy is mean $\pm$ sample standard deviation across ten splits after 200 attempted rounds. Full is the fraction retaining all workers; Workers is mean participation. The two partitions share failure sequences. Healthy communication uses no failures.

Both resilient supports preserve full participation and the healthy iterates. The accuracy gaps are small under these random failures, in contrast to the persistent, class-separated stress case in Appendix D. Nominal joint and terminalwise supports preserve all workers in 69.8% and 58.8% of attempted rounds. Consequently, high final accuracy alone does not establish preservation of the specified computation. The resilient supports satisfy that stronger requirement, while their identical parameter trajectories are the expected consequence of Corollary 8.

We also measure path rebuilds, message-hop counts, route-maintenance time, and sequential gradient-computation time in the simulator (Appendix H). Joint and pruned supports require means of 24.7 and 27.2 message hops per attempted round, versus 20.0 in the healthy reference. Routing is recomputed only for invalid cached paths. These measurements describe work and single-process simulation cost. They do not measure failure detection, network contention, or end-to-end distributed training time.

7 Limitations and open questions

The guarantee concerns small supports and a supplied narrow decomposition of their relevant neighborhood. It does not assume $\ell = o(N)$ in every application. A spanning deployment already needs at least $|V| - 1$ arcs, and larger survivability demands force larger budgets. When ℓ or the realized width is large, the exponential factor can dominate. The direct comparison measures that limitation even on narrow grids, including cases where the flow state improves on path pairings. The separate provisioning study uses a mixed-integer solver and does not establish an application-scale advantage for the DP.

Local treewidth is a substantive restriction. Low average degree, geographically nearby devices, and a directed physical underlay do not establish it. The generic proof also does not yield a polynomial

kernel or an ETH-tight exponent. Unrestricted W[1]-hardness rules out an FPT algorithm in ℓ on all digraphs unless $\mathrm{FPT} = \mathrm{W}[1]$, but supplies no matching lower bound for our local-width dependence. Neither the demand-profile dependence nor the state count is claimed optimal.

Finally, provisioning survivability does not control congestion, failure-detection overhead, deadlines, or learning under corrupt messages. Execution preservation requires stable per-round failures, reliable delivery, and completion of the prescribed participation set. Combining a survivability certificate with measured communication capacity and realistic reconfiguration costs is the next systems question. The present result isolates a design primitive and makes the conditions under which it preserves the learning computation explicit.

AI USE STATEMENT

Generative AI assisted in literature search, grammatical and structure revision, and figure construction. The authors assume responsibility for the content.

REFERENCES

E. Alpaydin and C. Kaynak. Optical recognition of handwritten digits. UCI Machine Learning Repository, 1998. URL https://archive.ics.uci.edu/dataset/80/optical+recognition+of+handwritten+digits.

Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12(2):308–340, 1991. doi: 10.1016/0196-6774(91)90006-K.

Hans L. Bodlaender, Pål G. Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lokshtanov, and Michał Pilipczuk. A $c^k n$ 5-approximation algorithm for treewidth. *SIAM Journal on Computing*, 45(2): 317–378, 2016. doi: 10.1137/130947374. URL https://arxiv.org/abs/1304.6321.

Parinya Chalermsook, Syamantak Das, Guy Even, Bundit Laekhanukit, and Daniel Vaz. Survivable network design for group connectivity in low-treewidth graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 116 of *LIPIcs*, pp. 8:1–8:19, 2018. doi: 10.4230/LIPIcs.APPROX-RANDOM.2018.8. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.APPROX-RANDOM.2018.8.

Chun-Hsiang Chan, Bundit Laekhanukit, Hao-Ting Wei, and Yuhao Zhang. Polylogarithmic approximation algorithm for k-connected directed Steiner tree on quasi-bipartite graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 176 of *LIPIcs*, pp. 63:1–63:20, 2020. doi: 10.4230/LIPIcs.APPROX/RANDOM.2020.63.

Erik D. Demaine and MohammadTaghi Hajiaghayi. Equivalence of local treewidth and linear local treewidth and its algorithmic applications. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 833–842, 2004. URL https://erikdemaine.org/papers/ApexMinorFree_SODA2004/.

Andreas Emil Feldmann, Anish Mukherjee, and Erik Jan van Leeuwen. The parameterized complexity of the survivable network design problem. *Journal of Computer and System Sciences*, 148:103604, 2025. doi: 10.1016/j.jcss.2024.103604. URL https://arxiv.org/abs/2111.02295.

Fabrizio Grandoni and Bundit Laekhanukit. Surviving in directed graphs: A quasi-polynomial-time polylogarithmic approximation for two-connected directed Steiner tree. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 420–428, 2017. doi: 10.1145/3055399.3055445.

Mark Jones, Daniel Lokshtanov, M. S. Ramanujan, Saket Saurabh, and Ondřej Suchý. Parameterized complexity of directed Steiner tree on sparse graphs. *SIAM Journal on Discrete Mathematics*, 31 (2):1294–1327, 2017. doi: 10.1137/15M103618X. URL https://arxiv.org/abs/1210.0260.

Othmane Marfoq, Chuan Xu, Giovanni Neglia, and Richard Vidal. Throughput-optimal topology design for cross-silo federated learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e29b722e35040b88678e25a1ec032a21-Abstract.html.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282, 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

Runze You and Shi Pu. B-ary tree push-pull method is provably efficient for distributed learning on heterogeneous data. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/b0d4b9d3dc41b36cad1d36ddd27858ab-Abstract-Conference.html.

A FAILURE EQUIVALENCE AND JOINT OPTIMIZATION

For completeness, the directed failure interpretation is exact for a fixed support F. Fix distinct vertices s, t and an integer $b \geq 1$. The following statements are equivalent.

1. $D[F]$ contains b pairwise arc-disjoint directed s–t paths.
2. For every $S \subseteq F$ with $|S| \leq b-1$, the graph $D[F \setminus S]$ contains an s–t path.
3. Every directed cut separating s from t contains at least b arcs of F directed from the source side to the sink side.

The first statement implies the second because one deleted arc can intersect at most one of the chosen paths. The second implies the third because deleting a smaller cut destroys all such paths. For the last implication, put unit capacities on selected arcs and augment an integral flow along residual paths until no augmenting path exists. At termination, residual reachability defines a cut whose capacity equals the integral flow value. If every separating cut has capacity at least b, this value is at least b. Decompose the integral flow into paths and cycles, discard cycles, and retain b simple paths. Unit capacities make those paths arc-disjoint. This argument also applies to distinct parallel arcs.

Applying the equivalence separately to every worker and direction proves the failure interpretation of Definition 1. For a common failure set of size at most f, setting every $b_t^+, b_t^- \geq f+1$ guarantees both routes for every worker simultaneously. The paths themselves need not be disjoint across workers or opposite directions, because reachability and subsequent scheduling are separate requirements.

Heterogeneous failure guarantees. If demands differ, connection (r, t) tolerates any $b_t^+ - 1$ deletions, and (t, r) tolerates any $b_t^- - 1$ deletions. Full participation under a common failure set follows whenever its size is at most $\min_t\{b_t^+ - 1, b_t^- - 1\}$. Larger individual demands can protect selected workers more strongly, but do not improve this all-worker bound beyond the least protected direction.

Why two independent optima do not give the joint optimum. Let F^+ and F^- minimize outward and inward costs separately. Their union is feasible for the directional connectivity requirements, but may exceed a shared budget. Even without a budget, its cost need not be minimum, because an alternative outward support can share more arcs with an inward support. The objective $c(F^+ \cup F^-)$ is not generally $c(F^+) + c(F^-)$. The flow DP represents this overlap at the arc-introduction transition and imposes the budget on the union.

B DECOMPOSITION AND BOUNDARY CASES

Constructing a nice decomposition. Both dynamic programs use an elimination order from a deterministic minimum-fill heuristic. Each eliminated vertex and its still-active neighbors form a bag. Fill edges are used only to construct the decomposition. They are never available as directed routing arcs. Connect each bag to the bag of its earliest remaining neighbor, and join disconnected components through an empty bag. Recursively introduce or forget vertices to make adjacent bags differ by one vertex and combine children through equal-bag joins. Assign an original arc to the bag of its earlier-eliminated endpoint and insert its unique introduce-arc node there. Both original endpoints occur in that bag. This procedure produces a valid decomposition of the original underlying graph, with a measured width. The heuristic does not certify minimum treewidth or by itself implement the graph-class decomposition guarantee.

Flow states and reconstruction. The flow solver uses one integer imbalance per requirement and bag vertex. An arc has a binary usage decision in each requirement, regardless of its multiplicity. At a forget node, divergence must be b_g at a source, $-b_g$ at a sink, and zero otherwise. Circulations are allowed. Nonnegative costs and the bounded-witness lemma justify this choice without storing components or enforcing acyclicity. The labelled-path solver provides a baseline. It stores boundary degrees and ordered segment pairings for each path copy, tests directed cycles, and keeps distinct markers for forgotten sources and sinks. Both solvers use exactly the same deterministic nice decomposition on each input.

Boundary cases. Empty worker sets return the empty support. A positive demand with $\ell = 0$ is infeasible. An isolated required endpoint is rejected when its forget transition fails the divergence rule. A required terminal outside $B_\ell(r)$ is rejected before decomposition. Zero-cost arcs cause no problem because minimizing cost still permits taking an inclusion-minimal optimal support, and the explicit cardinality counter distinguishes budgets. Parallel arcs have independent identifiers, costs, and failure events. Demands exceeding the directional arc budget imply infeasibility before any flow coordinates are allocated. These conventions avoid hiding dependence on large binary-encoded demands or on an omitted multiarc count.

State minimization and reconstruction. For each signature and union cardinality, the table retains a minimum-cost realization and its selected arcs. Future compatibility depends only on the signature and counter, so a more expensive realization can be discarded. Predecessor pointers offer an equivalent way to recover the selected support. Integer costs permit integer arithmetic throughout. For rational costs, additions and comparisons have polynomial bit complexity. Localization applies before this optimization.

C Independent flow formulation

Let $\mathcal{R}$ be the set of ordered requirements, with $h = (s_h, t_h, b_h)$. Introduce a binary provisioning variable x_e for each original arc and a continuous flow $y_{h,e}$ for each requirement and arc. The formulation is

$$\min \quad \sum_{e \in A} c_e x_e, \tag{8}$$

$$\text{subject to} \quad \sum_{e \in A} x_e \le \ell, \tag{9}$$

$$\sum_{e \in \delta^+(v)} y_{h,e} - \sum_{e \in \delta^-(v)} y_{h,e} = b_h\big(\mathbf{1}\{v = s_h\} - \mathbf{1}\{v = t_h\}\big) \quad (h \in \mathcal{R}, v \in V), \tag{10}$$

$$0 \le y_{h,e} \le x_e \quad (h \in \mathcal{R}, e \in A), \tag{11}$$

$$x_e \in \{0,1\} \quad (e \in A). \tag{12}$$

For a fixed binary support, feasibility of a value-b_h unit-capacity flow implies an integral flow of that value and hence the required disjoint paths. Thus the continuous flow variables are sufficient. Capacities are imposed separately for every requirement, matching the survivability semantics. Summing $y_{h,e}$ across requirements and bounding that sum by x_e would impose an unintended shared-capacity constraint and solve a different problem.

The provisioning experiments use SciPy/HiGHS with a zero requested relative gap and a 60-second solve limit. Every reported design attains a zero solver gap and satisfies the original connectivity and cardinality constraints. These are numerical optimization results, with the small-instance enumeration providing an independent comparison. Further details are in the supplement.

Replacing c_e by one finds the minimum feasible cardinality. Independent routing uses one requirement at a time without a binding global budget. Separate-direction routing solves the three outward or inward requirements under the displayed budget and unions the two solutions. Nominal joint routing uses $b_h = 1$ for all six requirements. Nominal terminalwise routing unions six separately optimal one-path supports. No baseline is credited with meeting the combined budget when its union exceeds it.

D Experimental specification and full results

Graph generation and budgets. Grid vertices have row-major identifiers $iw + j$ for $0 \le i < h$ and $0 \le j < w$. Insert both arc orientations for every horizontal or vertical neighboring pair, preserving them as separate directed resources. Arc costs are independent uniform draws from $\{1, \dots, 10\}$ using NumPy's default random generator and the stated seed. The root is vertex zero and the workers are $w - 1$, $(h - 1)w$, and $hw - 1$. The sizes contain respectively 12, 20, and 30 vertices and 34, 62, and 98 arcs. The minimum bidirectional two-path cardinalities are 20, 28, and 36, so the studied budgets

Grid	Method	Mean cost	Mean arcs	Overbudget	Full fraction
3×4	Joint resilient	121.8	20.2	0/10	1.000
3×4	Independent resilient	168.1	30.3	10/10	1.000
3×4	Separate directions	148.7	27.7	10/10	1.000
3×4	Joint nominal	54.8	11.4	0/10	0.665
3×4	Terminalwise nominal	83.2	16.7	0/10	0.559
4×5	Joint resilient	157.4	29.2	0/10	1.000
4×5	Independent resilient	236.0	45.3	10/10	1.000
4×5	Separate directions	188.0	38.0	10/10	1.000
4×5	Joint nominal	72.1	16.4	0/10	0.735
4×5	Terminalwise nominal	106.5	24.6	0/10	0.650
5×6	Joint resilient	197.4	38.8	0/10	1.000
5×6	Independent resilient	294.9	59.6	10/10	1.000
5×6	Separate directions	239.6	51.1	10/10	1.000
5×6	Joint nominal	88.6	21.0	0/10	0.786
5×6	Terminalwise nominal	131.5	31.1	0/10	0.710

Table 4: Support costs and diagnostics across ten cost seeds per grid. Overbudget unions are not feasible alternatives under the common cardinality bound. Full-round fractions enumerate deletion of each original underlay arc, including arcs outside the chosen support.

Grid	Feasible trials	Joint cost	Pruned cost
3×4	10/10	121.8	127.2
4×5	3/10	157.0	168.0
5×6	1/10	185.0	208.0

Table 5: Deterministic pruning of the independent resilient union. Costs are means over only those trials where pruning meets the common budget; joint costs use the same trials. Every pruned support meets the connectivity requirements, including those that remain over budget.

are 24, 32, and 40. These budgets are chosen through a separate minimum-cardinality solve and are held common across compared designs. The underlays are synthetic planar grids. Their shape is not inferred from a real communication deployment.

Budget-aware pruning baseline. Start from the independent resilient union. Consider its arcs once in descending provisioning-cost order, breaking ties by increasing arc identifier. Delete an arc whenever all six two-path requirements remain feasible. Further deletions cannot make an earlier rejected deletion feasible, so one pass produces an inclusion-minimal support. This procedure preserves connectivity, but need not meet the common cardinality budget. No random restarts or tuned scoring parameters are used. Table 5 gives the feasibility rate and compares costs on matched feasible trials. The baseline is a simple pruning heuristic, not a claimed approximation algorithm for the budgeted problem.

Failure enumeration. For each selected support, remove each original underlay arc in turn, including arcs absent from that support. A worker remains available only if it is reachable from the root and can reach the root in the remaining directed support. The full-round fraction is the number of deletions retaining all three workers divided by the total number of original underlay arcs. It is an exhaustive fraction under a uniform enumeration convention. In particular, it is not conditional on hitting a selected link, and not an empirical reliability estimate. The resilient solutions have fraction one under either selected-only or all-underlay enumeration.

Learning data and objective. The learning data are the 1,797 examples in scikit-learn's bundled digits dataset, a subset of the 5,620-example UCI distribution. Scale the 64 integer pixels by $1/16$ and append a constant feature, giving $X \in \mathbb{R}^{1797 \times 65}$. Each of ten stratified splits has 1,347 training and 450 test examples. Worker $j \in \{0, 1, 2\}$ receives training examples with digit label congruent to j modulo three. Let n_j be its sample count and $w_j = n_j/1347$. For $W \in \mathbb{R}^{65 \times 10}$, minimize the training cross-entropy plus $0.001\|W\|_F^2/2$, including the constant-feature row in the regularizer.

Initialize $W = 0$ and use 200 full-batch gradient steps of size 0.5. Test loss excludes the regularizer. There is no validation-based tuning, early stopping, or selection of a preferred data split.

Partial-outage stress rule. The network is always the 3×4 instance with cost seed zero. The nominal terminalwise support has 15 arcs and cost 80. The joint resilient support has 20 arcs and cost 119, within budget 24. The independent resilient support has 31 arcs and cost 177, exceeding that budget. Among all single-arc deletions leaving at least one but fewer than three workers available in the nominal terminalwise support, choose the deletion minimizing the sum of available training weights, breaking ties by arc identifier. On these splits this deletes the arc $0 \to 1$ and retains only the worker holding labels congruent to one modulo three. The rule is evaluated from connectivity and training sample counts, never from test labels or outcomes. Hold that same deletion fixed for every round and every support in the comparison.

The simulator recomputes directed reachability after the deletion and assumes exact message delivery along surviving routes. It evaluates the prescribed local gradients at the received common model, and the coordinator combines them in fixed worker order. If some workers are unavailable in the nominal support, it renormalizes the surviving dataset-size weights. A strict full-participation protocol on that support would instead fail to complete the first round. The experiment uses the skip rule to expose the resulting change in the optimization population. It does not model packets, queueing, retransmission overhead, or a wall-clock network schedule.

Reported uncertainty and interpretation. Reported spreads are sample standard deviations across the ten specified random train/test splits of the same dataset. These are dependent samples from a shared finite dataset and are not confidence intervals for deployments or populations. Mean final test cross-entropy is 0.30593 for healthy, joint resilient, and independent resilient execution, versus 5.11394 for nominal execution with missing workers. Each resilient parameter trajectory coincides with the healthy trajectory, as predicted by Corollary 8. This identity does not establish a new convergence law. The large accuracy difference is specific to the persistent failure and the intentionally separated class distributions.

Small-instance comparisons. The first comparison contains 80 random cases on 3–5 vertices and at most 11 arcs, with integer costs from zero to seven, and eight targeted cases. The latter cover joins, cycles, parallel arcs, sharing, opposite directions, unreachable terminals, a zero budget, and empty requirements. Both DPs, using the same nice decompositions, agree with exhaustive enumeration and MILP on all 88 optima. Seventeen cases contain join nodes and 55 are feasible. A further 100 seeded multigraphs use 3–5 vertices, 3–10 arcs drawn with replacement, costs from zero to three, and six requirement patterns, including demand three and a nonrooted pair list. Across every budget from zero to the number of arcs, all 742 flow–enumeration comparisons agree. Of these comparisons, 286 contain joins. The supplement gives further details.

E SECONDARY BASELINES AND THE NOVELTY BOUNDARY

The following observation is standard and is included only to clarify the baselines. Without a cardinality constraint, independently computing a minimum-cost unit-capacity flow for each of q requirements and taking their union gives a q-approximation to the nonuniform survivable union-cost problem. Each separately optimal requirement costs at most the cost of a joint optimum, and the cost of the union is at most the sum of those costs. The union is feasible because it contains the witness for every requirement. A cardinality bound invalidates the feasibility conclusion for the budgeted problem. Therefore this observation does not supply a q-approximation to Definition 1 under its original budget.

The corresponding undirected vertex-connectivity baselines and the known local-treewidth theorem for VC-SNDP are not contributions of this paper. The latter is due to Feldmann et al. (2025). Its undirected vertex-connectivity setting is distinct from the directed shared-cost problem studied here.

What is not implied by the hardness result. The general-graph W[1]-hardness for two-connected directed Steiner design does not establish the optimality of the exponent in (2). It provides no ETH-tight dependence on the local width or on the solution budget within planar classes. The

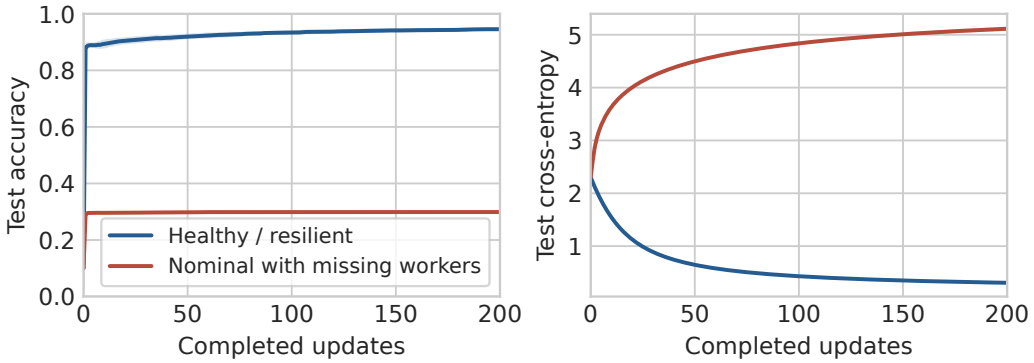

Figure 1: Persistent partial-outage mechanism example from the class-separated partition. Means and sample standard deviations cover ten data splits. The failed arc is chosen by the stated topology-based stress rule. Resilient and healthy parameter trajectories coincide; the nominal terminalwise design trains on one surviving worker. The independent resilient support in this earlier comparison exceeds the shared budget. The random-failure study uses only budget-feasible alternatives.

bounded-imbalance representation yields an upper bound, not a lower bound against further state compression. No matching complexity lower bound is claimed.

F Comparison with existing tractability results

Table 6 separates the graph model, demand structure, parameter, and constructive conclusion. The theorem numbers for Feldmann et al. (2025) follow the author preprint linked in the bibliography. The topological-minor result of Jones et al. (2017) appears as Theorem 10 in its linked preprint, Section 3.2. Those locators identify the statements being compared rather than attributing the present state construction to either paper.

Why qualitative tractability alone is not the novelty claim. The localized graph can be represented by an incidence structure with directed tail and head relations. With a bounded number of demands, one can quantify a path-arc set for each path copy, require directed source–sink connectivity, and enforce disjointness within each group. The selected support is the union of these sets. A budget of at most ℓ arcs can also be encoded with a formula whose length depends on ℓ. Nonnegative union costs give a logical optimization problem on a bounded-width structure. Together with localization, this is an established route to a qualitative FPT consequence. Theorem 2 replaces that route by an explicit boundary representation, transition rules, and state count. The bidirectional extension reuses the same generic flow DP; it is not claimed as an independent complexity breakthrough.

Connection-profile algorithms. Chalermsook et al. (2018, Theorem 4) give rooted edge- and vertex-connectivity algorithms with running time $f_2(k, w)n$, where k is maximum demand and w is treewidth. Their Section 4 uses local and global connectivity profiles, with at most $\exp(w^{O(wk)})$ valid cells per bag in Lemma 8. This is an important precedent for explicit bounded-width survivable-design algorithms. Our state has one bounded divergence per ordered requirement and boundary vertex, plus a union-cardinality counter. It gives the product in (6), with exponential dependence on the number of requirements. Accordingly, we neither claim the first bounded-width algorithm for rooted survivability nor assert that our bound uniformly dominates a bound independent of that number. The specific result is the stated demand-profile bound for independent directed arc-capacity certificates with shared provisioning and a cardinality budget.

Scope of the learning comparison. Marfoq et al. (2020) optimize a throughput objective on a learning underlay, whereas the present objective is offline provisioning cost under worst-case reachability and cardinality constraints. You & Pu (2024) analyze a particular decentralized algorithm on directed trees. Corollary 8 concerns a different, prescribed synchronous coordinator computation and

Result	Model and demands	Parameter and graph class	Constructive conclusion
Feldmann et al., Thm. 1.4	Undirected VC-SNDP; arbitrary terminal pairs and vertex-disjoint paths	Selected-edge count ℓ; locally bounded treewidth	$f(\ell)n$ algorithm through localization and logical optimization
Feldmann et al., Thm. 1.5	Same undirected pair-demand model	Treewidth plus total demand; unrestricted graph class	$f(\mathrm{tw}+D)n$ algorithm through an explicit logical formulation
Feldmann et al., Thm. 1.6	Rooted directed 2-DST; uniform outward demand two, arc disjointness	Selected-arc count ℓ; unrestricted digraphs	W[1]-hardness; no restricted-width exponent lower bound
Jones et al., topological-minor theorem	Rooted directed Steiner tree; unit outward reachability	Number of selected nonterminals; fixed excluded topological minor	Constructive FPT algorithm for that parameter and unit demand
Chalermsook et al., Thm. 4	Rooted edge- and vertex-connectivity design	Maximum demand k and treewidth w	$f_2(k,w)n$ algorithm with connectivity profiles, independent of requirement count
Arnborg et al., logical optimization	Definable bounded-treewidth optimization with annotated structures	Width and formula size	General qualitative route, with no claimed exponent matching the present DP
Theorem 2	Rooted directed multigraph; nonuniform outward and inward arc demands; shared union cost	Demand profile κ, ball width τ, budget ℓ; a given decomposition	Bounded-imbalance DP with the quantified running time in (2)

Table 6: Theorem-level comparison. Selected-edge, selected-arc, and selected-nonterminal parameters refer to different objects. Vertex-disjoint undirected paths and arc-disjoint directed paths are also different requirements.

does not transfer that algorithm's convergence theorem. A feasible support supplies a communication precondition, while routing, congestion, and wall-clock performance remain separate decisions.

G MATCHED SOLVER STUDY AND TRACTABILITY MAP

Retained suite. The original 40 cases cross grids of sizes 2×4, 2×10, 3×4, and 4×5, cost seeds zero and one, and five requirement patterns. With v the opposite corner, the first two patterns require respectively one and two r–v paths. The third requires two paths in each direction between r and v. The fourth and fifth require respectively one and two paths in each direction for all three other corners. Thus (q, p) equals $(1, 1)$, $(1, 2)$, $(2, 4)$, $(6, 6)$, and $(6, 12)$. All original arc lists, costs, requirements, and budgets are retained. The budget equals minimum feasible cardinality plus four, capped by the arc count.

Prespecified stress suite. Before running the comparison, we fix 36 additional cases. The underlying grid is 2×4 or 3×4, with two parallel copies of each directed arc and independently drawn integer costs from one to ten. Widths are two and three, and arc counts are 40 and 68. Requirement counts are $q \in \{1, 2, 6\}$. These connect the root to the opposite corner, both directions for that corner, or both directions for all three other corners, respectively. Every requirement has multiplicity $b \in \{1, 2, 3\}$. The budget is minimum feasible cardinality plus $s \in \{0, 4\}$. This full crossing varies the number of groups and multiplicities separately. The two demand profiles $(q, b) = (2, 3)$ and $(6, 1)$ have the same total demand six. They need not have the same minimum cardinality or geometric demand pattern, so the comparison is not an isolated causal experiment in q.

Computational cost. Every solver has a ten-second limit, and each DP table has a 50,000-state cap. The DPs use the same minimum-fill decomposition. Their solve times include decomposition construction, while MILP times include formulation and support evaluation. Imports and budget selection are excluded. Measurements use one BLAS/OpenMP thread on a shared machine. Neither

DP uses state compression or accelerated joins. Flow states permit some circulation-containing assignments that the acyclic path representation excludes, so the asymptotic bound does not order the realized state counts on every instance.

Transition counts include every locally feasible candidate, whether or not it improves a stored value. Join work counts all examined child-state pairs. Peak resident memory includes the interpreter and dependencies, so differences in process memory cannot be attributed solely to the algorithms. A resource limit yields neither an optimum nor an infeasibility conclusion. Additional results are in the supplement.

Table 7: Retained instances. Solver entries give seconds followed by status, with O for optimal, C for the state cap, and T for the time limit. Multiplicities are uniform within a row and $p = qb$. Seeds are zero for IDs 0–4, 10–14, 20–24, 30–34 and one for their successors.

ID	Grid	τ	m	q	b	ℓ	Flow	Pairing	MILP
0	2×4	2	20	1	1	8	< 0.001 O	< 0.001 O	0.004 O
1	2×4	2	20	1	2	12	< 0.001 O	0.001 O	0.003 O
2	2×4	2	20	2	2	20	0.003 O	0.017 O	0.003 O
3	2×4	2	20	6	1	12	0.528 C	0.582 C	0.005 O
4	2×4	2	20	6	2	20	0.093 C	1.417 C	0.003 O
5	2×4	2	20	1	1	8	< 0.001 O	< 0.001 O	0.004 O
6	2×4	2	20	1	2	12	< 0.001 O	0.001 O	0.004 O
7	2×4	2	20	2	2	20	0.003 O	0.017 O	0.003 O
8	2×4	2	20	6	1	12	0.557 C	0.629 C	0.005 O
9	2×4	2	20	6	2	20	0.098 C	1.396 C	0.003 O
10	2×10	2	56	1	1	14	0.002 O	0.002 O	0.004 O
11	2×10	2	56	1	2	24	0.002 O	0.007 O	0.003 O
12	2×10	2	56	2	2	44	0.017 O	0.173 O	0.005 O
13	2×10	2	56	6	1	24	0.544 C	0.639 C	0.012 O
14	2×10	2	56	6	2	44	0.096 C	1.410 C	0.006 O
15	2×10	2	56	1	1	14	0.002 O	0.002 O	0.007 O
16	2×10	2	56	1	2	24	0.002 O	0.008 O	0.005 O
17	2×10	2	56	2	2	44	0.017 O	0.175 O	0.003 O
18	2×10	2	56	6	1	24	0.558 C	0.610 C	0.009 O
19	2×10	2	56	6	2	44	0.094 C	1.364 C	0.007 O
20	3×4	3	34	1	1	9	0.002 O	0.002 O	0.004 O
21	3×4	3	34	1	2	14	0.001 O	0.010 O	0.004 O
22	3×4	3	34	2	2	24	0.027 O	2.292 C	0.005 O
23	3×4	3	34	6	1	14	0.265 C	2.100 C	0.007 O
24	3×4	3	34	6	2	24	0.104 C	0.460 C	0.006 O
25	3×4	3	34	1	1	9	0.002 O	0.002 O	0.004 O
26	3×4	3	34	1	2	14	0.002 O	0.010 O	0.004 O
27	3×4	3	34	2	2	24	0.030 O	2.183 C	0.004 O
28	3×4	3	34	6	1	14	0.272 C	1.972 C	0.007 O
29	3×4	3	34	6	2	24	0.104 C	0.433 C	0.005 O
30	4×5	4	62	1	1	11	0.008 O	0.011 O	0.005 O
31	4×5	4	62	1	2	18	0.014 O	0.484 O	0.005 O
32	4×5	4	62	2	2	32	0.754 C	0.572 C	0.005 O
33	4×5	4	62	6	1	18	0.555 C	0.506 C	0.009 O
34	4×5	4	62	6	2	32	0.095 C	0.324 C	0.010 O
35	4×5	4	62	1	1	11	0.008 O	0.009 O	0.004 O
36	4×5	4	62	1	2	18	0.015 O	0.486 O	0.004 O
37	4×5	4	62	2	2	32	0.782 C	0.589 C	0.005 O
38	4×5	4	62	6	1	18	0.577 C	0.496 C	0.015 O
39	4×5	4	62	6	2	32	0.108 C	0.327 C	0.009 O

Table 8: Prespecified stress instances. Solver entries give seconds followed by status, with O for optimal, C for the state cap, and T for the time limit. Multiplicities are uniform within a row and $p = qb$. Each grid uses one fixed cost draw. Within each consecutive pair of rows, budget slack is zero and four.

ID	Grid	τ	m	q	b	ℓ	Flow	Pairing	MILP
40	2×4	2	40	1	1	4	< 0.001 O	< 0.001 O	0.004 O
41	2×4	2	40	1	1	8	0.001 O	< 0.001 O	0.005 O
42	2×4	2	40	1	2	8	0.001 O	0.005 O	0.004 O
43	2×4	2	40	1	2	12	0.002 O	0.004 O	0.004 O
44	2×4	2	40	1	3	12	0.002 O	0.016 O	0.004 O
45	2×4	2	40	1	3	16	0.002 O	0.023 O	0.005 O
46	2×4	2	40	2	1	8	0.007 O	0.004 O	0.005 O
47	2×4	2	40	2	1	12	0.009 O	0.006 O	0.005 O
48	2×4	2	40	2	2	16	0.036 O	0.407 O	0.005 O
49	2×4	2	40	2	2	20	0.054 O	0.558 O	0.005 O
50	2×4	2	40	2	3	24	0.057 O	0.908 C	0.005 O
51	2×4	2	40	2	3	28	0.072 O	0.932 C	0.005 O
52	2×4	2	40	6	1	8	0.212 C	2.756 O	0.007 O
53	2×4	2	40	6	1	12	0.203 C	1.678 C	0.007 O
54	2×4	2	40	6	2	16	0.589 C	2.140 C	0.006 O
55	2×4	2	40	6	2	20	0.608 C	2.172 C	0.006 O
56	2×4	2	40	6	3	24	0.614 C	10.023 T	0.007 O
57	2×4	2	40	6	3	28	0.580 C	10.154 T	0.008 O
58	3×4	3	68	1	1	5	0.003 O	0.002 O	0.005 O
59	3×4	3	68	1	1	9	0.003 O	0.004 O	0.005 O
60	3×4	3	68	1	2	10	0.006 O	0.023 O	0.005 O
61	3×4	3	68	1	2	14	0.011 O	0.052 O	0.005 O
62	3×4	3	68	1	3	15	0.010 O	0.341 O	0.005 O
63	3×4	3	68	1	3	19	0.012 O	1.126 O	0.005 O
64	3×4	3	68	2	1	10	0.056 O	0.039 O	0.007 O
65	3×4	3	68	2	1	14	0.107 O	0.098 O	0.006 O
66	3×4	3	68	2	2	20	1.381 O	1.413 C	0.006 O
67	3×4	3	68	2	2	24	2.464 O	1.395 C	0.006 O
68	3×4	3	68	2	3	30	0.622 C	0.437 C	0.006 O
69	3×4	3	68	2	3	34	0.619 C	0.438 C	0.005 O
70	3×4	3	68	6	1	10	0.222 C	0.676 C	0.009 O
71	3×4	3	68	6	1	14	0.217 C	0.631 C	0.010 O
72	3×4	3	68	6	2	20	0.623 C	0.421 C	0.010 O
73	3×4	3	68	6	2	24	0.603 C	0.434 C	0.009 O
74	3×4	3	68	6	3	30	0.620 C	2.571 C	0.010 O
75	3×4	3	68	6	3	34	0.586 C	2.511 C	0.011 O

Suite	Method	Peak states	Trans. M	Join pairs M	Accepted M	RSS MiB
Original	Flow	50,001	3.65	0.18	0.16	37.5
Original	Pairing	50,001	2.19	0.73	0.20	144.3
Original	MILP	–	–	–	–	125.0
Stress	Flow	50,001	6.93	6.72	0.24	42.8
Stress	Pairing	50,001	4.24	0.18	0.02	57.2
Stress	MILP	–	–	–	–	123.9

Table 9: Work and memory including all incomplete runs. Counts in millions are summed over runs, while table size and process RSS are maxima. Peak tables can exceed the cap by one state. Further results are in the supplement. MILP does not expose comparable DP counters.

Every completed DP optimum equals the MILP optimum on the same instance. The observed tractability map describes these graph and demand families. It does not establish a universal threshold in width, demand, or budget.

Grid	Joint / underlay	Joint / selected	Terminal / underlay	Terminal / selected
3×4	66.5	0.0	55.9	10.3
4×5	73.5	0.0	65.0	11.9
5×6	78.6	0.0	71.0	8.7

Table 10: Full-round retention (percent) for nominal designs, averaged over ten cost seeds. Underlay samples all input arcs equally; selected conditions on deleting a provisioned arc and therefore changes the sampling set with the design. Both resilient methods retain every worker under either convention whenever the pruning baseline is budget-feasible.

H Random-failure learning and routing measurements

Designs and failure distributions. The learning comparison uses the same fixed 3×4 cost-seed-zero instance as the mechanism example, with common budget 24. Joint resilient and reverse-delete pruned supports have respectively 20 and 24 arcs and costs 119 and 144. Nominal joint and terminalwise supports have 10 and 15 arcs and costs 59 and 80. All four alternatives satisfy the common cardinality budget, though the nominal alternatives require only one path per direction and therefore do not satisfy the resilient connectivity specification. They measure the consequence of relaxing survivability, rather than competing as feasible solutions to the two-path optimization problem.

For the full set of 30 graph instances, we enumerate both a uniform deletion from the entire underlay and a uniform deletion conditional on membership in the selected support. The latter changes its sampling set with the design and is reported as conditional vulnerability. Neither convention is fitted to empirical failure rates. Parallel logical channels sharing a physical failure domain would need to be deleted together, which is outside these single-arc experiments.

Data allocation and update rule. Use the same ten stratified train/test splits, preprocessing, objective, initialization, step size, and 200-step horizon as the mechanism example. The balanced allocation shuffles each class within the training split and divides it as evenly as possible among three workers. For heterogeneous allocations, independently draw each class's worker proportions from $\mathrm{Dirichlet}(0.5, 0.5, 0.5)$ and use a multinomial allocation of that class's training examples. Resample only if a worker would have no examples. Worker weights are their fractions of the complete training set. Partition randomness uses seeds $7000 + s$ for split s.

For each split, seeds $17000 + s$ draw a sequence of 200 uniform underlay-arc deletions, one per round. A failure remains fixed within the round and is restored before the next deletion. The same sequence is used across supports and across partition schemes. Failure selection is independent of the data allocation and test performance. If at least one worker remains available in both directions, the coordinator applies the sample-weighted gradient of the surviving population, with weights renormalized to sum to one. Otherwise no update is made. The horizon is 200 *attempted* rounds, so different supports can complete different numbers of updates. A strict full-participation protocol would instead stop on any incomplete round.

Routing and timing. Initialize six shortest directed paths, one per worker and direction, using breadth-first search with increasing arc identifiers to break ties. Retain a cached path while it remains valid. After a failure, recompute a path if it contains the failed arc or if no route was available in the previous round. A worker participates only if both cached directions are available. The message-hop count sums the outward and return path lengths for participating workers, assuming one message in each direction and sequential unicast. It excludes failed attempts, control traffic, and failure detection.

The measured routing time covers cache inspection and path rebuilding. Round time adds the actual sequential local-gradient calculations and coordinator aggregation. The 200-round total is the sum of these measured operations, with test evaluation, initial route construction, and data preparation excluded. It is computation time within a single-process simulator, not elapsed time for a deployed distributed job. In an ideal serialized network with equal per-hop message time Δ, the measured hop count would contribute $H\Delta$ per round in addition to detection and computation. No value of Δ or zero-overhead detection is inferred from the experiment.

Support	Hops	Rebuilds	Routing μs	Round ms	200 rounds s
Healthy	20.0	0.00	2.0	0.349	0.070
Joint resilient	24.7	0.73	5.8	0.352	0.070
Pruned resilient	27.2	0.80	6.4	0.352	0.070
Nominal joint	20.9	1.64	6.6	0.248	0.050
Nominal terminalwise	16.1	1.10	6.3	0.291	0.058

Table 11: Routing work and single-process simulation time for heterogeneous partitions. Means cover ten runs of 200 attempted rounds. Round time includes route maintenance and sequential gradient computation, excluding test evaluation and initialization. These are measured simulator times, not distributed training times. Hops count selected successful-worker messages.

Interpretation. Resilient executions preserve the healthy parameter arrays. Nominal joint failures either retain all workers or stop the update, whereas nominal terminalwise routing changes participation within a round. The modest accuracy differences in Table 3 contrast with the persistent, class-separated stress case, even though the prescribed computation is still altered.