Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Structure-Aware Exact Priority Optimization for Real-Time Cause–Effect Chains under Correlated Execution Demands

Haojie Cai

chatjie@163.com

Nanjing University

Hao Cai

Henan Polytechnic University

Wenlong Hu

Huazhong University of Science and Technology

Zhiwen Li

Nanjing University

Research Article

Keywords: real-time scheduling, cause–effect chains, static priorities, distributionally robust optimization, probability graph projection, binary decision diagrams

Posted Date: November 27th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11869370/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Structure-Aware Exact Priority Optimization for Real-Time Cause–Effect Chains under Correlated Execution Demands

Haojie Cai[1*], Hao Cai[2], Wenlong Hu[3], Zhiwen Li[1]

[1*]Nanjing University, 163 Xianlin Road, Nanjing, 210023, Jiangsu, China
ORCID (Haojie Cai): 0009-0002-4265-4662.
[2]Henan Polytechnic University, 2001 Century Avenue, Jiaozuo, 454003, Henan, China.
[3]Huazhong University of Science and Technology, 1037 Luoyu Road, Wuhan, 430074, Hubei, China.

*Corresponding author(s). E-mail(s): chatjie@163.com;

Abstract

Fixed priorities change causal-chain failure risks despite hard deadlines. We minimize worst-case source-label failure probability over hard-feasible priorities in finite preemptive uniprocessor job sets. Fair binary demands and selected-pair disagreement bounds define a common ambiguity set. A constraint edge reverses the unique optimum over a continuous interval in a nine-job instance. We derive exact structured-event risks and project nonessential low-degree variables while preserving feasible joint laws on retained variables. A finite-witness optimizer reuses attaining distributions to bound competing priorities and shares scenarios to certify optimality. Across 24 prespecified synthetic conditions, results match independent exact references. Against synchronous multi-priority compilation with witness transfer, geometric-mean paired-median runtime ratios are 0.410 (20 jointly solved) and 0.736 (14 nonzero-risk cases). Four-arm ablation ratios are 0.819 overall and 0.924 for nonzero-risk conditions versus active witness search without projection or closed forms; constant and single-bit shortcuts explain most gains, long chains show speedups are not universal.

Keywords: real-time scheduling, cause–effect chains, static priorities, distributionally robust optimization, probability graph projection, binary decision diagrams

1 Introduction

Meeting individual job deadlines does not imply that a designated observation reaches the end of a cause–effect chain by its deadline. In a preemptive system with first-dispatch reads and completion-time writes, a consumer can read a stale label before its producer completes, even though both jobs meet their deadlines. The priority order changes not only completion times but also whether and when the designated source label is propagated. Reaction-time and data-age analyses account for such end-to-end timing phenomena (Dürr et al. 2019; Günzel et al. 2025); packet-lineage semantics provide a related model of data propagation Ahmad et al. (2026).

Execution demands need not be independent. For example, several jobs may process data affected by a common input condition. When only partial dependence information is available, selecting a priority by its failure probability under a single assumed joint law can obscure important alternatives. The admissible joint distributions form a polytope: each job's demand bit is equally likely to be zero or one, and selected pairs have upper bounds on their probability of disagreement. Each hard-feasible priority order is evaluated under the joint distribution that maximizes its failure probability, and the same ambiguity set is used for every candidate. This permits a precise question: can a change in statistical dependence constraints, without any change in the job model or communication semantics, change the optimal static priority?

The key difficulty is that scheduling determines a priority-specific Boolean failure event, while dependence constraints determine which parts of that event can receive probability mass. A compact event may omit many demand variables, but those variables can still restrict the joint law of the essential variables. Neither dropping all variables outside the event's support nor counting the failing assignments suffices to determine its worst-case probability in general. The problem also requires comparing a potentially factorial number of fixed-priority assignments.

We make three contributions.

1. We characterize a hard-feasible nine-job instance whose four-stage cause–effect chain revisits a task. Adding one statistical constraint edge reverses the unique global optimum over a continuous interval of the disagreement cap. The risk functions for all six priority orders show how the added edge can lower the worst-case probability of some failure events while leaving others unchanged.
2. We characterize exact risk and invariance for selected scheduling-generated Boolean events and apply a constructive low-degree marginal projection rule to arbitrary events. Specializing classical coupling and measure-extension arguments, the rule preserves the entire feasible set of joint distributions on the retained variables and extends a maximizing distribution to the original demand space. A topology-invariance criterion and explicit counterexamples identify when graph simplification is sound and when it is not.
3. For each priority order it certifies, the risk solver uses a closed form or a flow problem, reduced when possible, to return the exact risk and a maximizing distribution on the original demand space. The outer optimizer reuses these distributions and

shares scenario evaluations across orders until its lower and upper bounds agree. We compare the solver with alternatives for the same objective and use a matched factorial ablation to distinguish the effects of projection and closed-form risk evaluation, including on long chains with no structural reduction.

The focus is an offline finite scheduling model, not an online adaptive scheduler. We use synthesized task sets and explicitly specified probability constraints. The strict reversal is an existence and structural result; its frequency in operational workloads is not established.

2 Related Work

Classical fixed-priority scheduling distinguishes job feasibility from the choice of priority assignment (Liu and Layland 1973; Audsley 2001). End-to-end timing analyses address reaction time and data age (Dürr et al. 2019; Günzel et al. 2023a, 2025). Choi et al. (2020) study fixed-priority scheduling of loosely dependent tasks; Tang et al. (2023) compare communication paradigms and their associated priority assignments; Paladino et al. (2024) optimize priorities under logical execution time (LET) communication. Ahmad et al. (2026) examine packet-based latency and schedulability in multirate directed acyclic graphs (DAGs). Two recent JSA studies address complementary scheduling and timing problems in automotive real-time systems. Wang et al. (2026) propose a three-stage static scheduling method for multi-rate AUTOSAR software component DAGs, addressing multi-core task assignment and job execution ordering to ensure schedulability and end-to-end latency. Ren et al. (2025) develop the δ-Bound method for maximum reaction-time analysis of periodic task chains under partitioned fixed-priority preemptive scheduling on multiprocessors; the method transforms the analysis into upper bounds on release intervals between successive data-propagation instances. These studies complement packet-oriented data-latency analysis by addressing static schedule construction and deterministic timing analysis. In contrast, the present work selects a strict static task-priority order for a finite uniprocessor job set and minimizes the worst-case probability of a designated source-label propagation failure over a common ambiguity set of partially specified execution-demand dependence. In our model, a task may appear several times in a chain, with a separate logical output register for each occurrence. Jobs read at first dispatch and write at completion. Failure occurs if the sink has not written the designated source label by the chain deadline.

Probabilistic fixed-priority assignment predates this work (Maxim et al. 2011). Probabilistic reaction-time analysis (Günzel et al. 2023b) and probabilistic end-to-end latency analysis and optimization (Lee et al. 2022; Han and Kim 2023) address closely related questions. Han and Kim (2023) first construct a task-to-core mapping under multicore partitioned earliest-deadline-first (EDF) scheduling, then optimize release phases using probabilistic end-to-end latency analysis (PEELA) and the nondominated sorting genetic algorithm II (NSGA-II) for multiobjective search. The probabilistic guarantee assumes independent execution times. Ben-Amor et al. (Ben-Amor et

al. 2020) study partitioned fixed-priority DAGs with independent probabilistic execution times and propose subtask-priority assignment heuristics. In contrast, our decision variable is a strict, deterministic uniprocessor task-priority order, and the joint demand law ranges over a common statistical ambiguity set. Correlation-tolerant analysis of job deadline-failure probability and distribution-agnostic analysis of periodic tasks (Marković et al. 2023, 2024) show the importance of dependence information. Markov-chain demand models also permit efficient probability bounds (Friebe et al. 2024). Here, the common ambiguity set is defined by marginal fairness and graph-indexed pairwise disagreement caps; our goal is exact optimization of the causal label failure event, rather than solely a job response-time probability bound.

Binary decision diagrams (BDDs) represent Boolean functions (Bryant 1986), and family-wise symbolic verification is an established method of sharing state exploration (Classen et al. 2010). Distributionally robust optimization with marginals (Chen et al. 2022), general distributionally robust optimization (Kuhn et al. 2025), distributionally robust fixed-interval scheduling (Branda 2018), and network-flow representations of robust binary optimization (Bodur et al. 2026) provide mathematical foundations. Atom-based sharp bounds on Boolean-event probabilities date to Hailperin (1965), and master–separation methods are established in robust optimization (Zeng and Zhao 2013). For fair Bernoulli variables the vector of pairwise disagreements is a cut-polytope point (Huber and Marić 2019). Edge subdivision and cycle inequalities in cut polyhedra (Barahona and Mahjoub 1986), together with consistent marginal extension (Vorob'ev 1962), provide the background for the local projection rule. The cut-coordinate description alone does not specify a retained higher-order joint law; the explicit extension kernel in Section 5 preserves that law. Tight Bernoulli sum bounds with given tree marginals, without a conditional-independence assumption, are studied by Padmanabhan and Natarajan (2021). Our tree OR result uses a selected terminal subset and one-sided disagreement caps, with a direct attaining law. We use these established mathematical tools to explain and certify how changes in dependence constraints affect the optimal fixed-priority order.

3 Model and Optimization Objective

3.1 Finite Uniprocessor Job Semantics

Let $\mathcal{T}$ be a finite set of tasks and $\mathcal{J} = \{j_0, \ldots, j_{n-1}\}$ a finite set of jobs. Time is measured in integer ticks. Job j_i has task $\tau_i \in \mathcal{T}$, release $r_i \in \mathbb{Z}_{\geq 0}$, relative hard deadline $d_i \in \mathbb{Z}_{>0}$, and integer execution requirements $c_i^{\mathrm{lo}}, c_i^{\mathrm{hi}} \in \mathbb{Z}_{>0}$ with $c_i^{\mathrm{lo}} \leq c_i^{\mathrm{hi}}$. The bit $X_i \in \{0, 1\}$ selects $c_i(0) = c_i^{\mathrm{lo}}$ or $c_i(1) = c_i^{\mathrm{hi}}$. Let π be a strict priority order on tasks. All jobs of a task have that task's priority, with ties within the task resolved by release time and then by job index. At integer boundary t, writes from jobs completing at t are committed before new releases are admitted. If any job is ready, the highest-priority ready job runs during $[t, t+1)$. The dispatched job snapshots its input labels only on its first dispatch, if it serves a chain occurrence. A higher-priority job released at boundary t is eligible for dispatch at t. A job completes, writing all stored occurrence labels, at the end of its final assigned tick. When no job is ready,

the processor idles for one tick. There is no migration, lock blocking, self-suspension, execution abort, or multiprocessor interference.

For a complete demand vector $x \in \{0,1\}^n$, the above rules determine a unique schedule and job completion times $C_i(\pi, x)$. The set of hard-feasible priority orders is

$$\Pi_{\text{hard}} = \{\pi : \ C_i(\pi, \mathbf{1}) \leq r_i + d_i \text{ for every } j_i \in \mathcal{J}\}. \tag{1}$$

Here $\mathbf{1}$ is the all-high-demand vector. Checking that vector is sufficient for *job* hard deadlines in this finite preemptive, predictable model; it is not sufficient for the causal-chain deadline condition below. For each priority order, the optimizer checks every job's deadline in the all-high schedule before compiling the failure event. It rejects the order if any job misses its deadline. If no hard-feasible order remains, it returns HardInfeasible. This screening relies on fixed demand-independent releases, work-conserving preemptive dispatch and the specified absence of suspension and blocking; successful propagation of the source label still requires separate analysis.

Lemma 1 (Fixed-priority predictability of job completion) *In the finite model above, for any fixed task-priority order π and any demand assignment $x \in \{0,1\}^n$, every job satisfies $C_i(\pi, x) \leq C_i(\pi, \mathbf{1})$. Thus Equation* (1) *suffices to check the hard job deadlines for all demand assignments. The chain deadline requires a separate check.*

Proof Consider tasks in descending fixed-priority order. Under identical release times, reducing the execution requirements of the highest-priority task cannot delay any of its jobs under the stipulated deterministic same-task queue discipline. Inductively, tasks below a fixed priority rank see no more interference from higher-priority work when any higher-priority demands are reduced; reductions in their own jobs cannot delay their completions either. Lower-priority work cannot interfere with higher-priority jobs. Hence each job completes no later under x than under the componentwise maximum demands. This is an instance of the established predictability property of priority-driven preemptive scheduling (Ha and Liu 1994). □

3.2 First-Read/Completion-Write Propagation

A chain is an *ordered list of task occurrences* $\gamma = (\tau^{(1)}, \ldots, \tau^{(k)})$; its task identities need *not* be distinct (e.g., $A \to B \to C \to B$). Each occurrence index $h \in \{1, \ldots, k\}$ has its own initially zero *logical output register*, even when task identities repeat. These are model channels, not hardware registers. From $t_0 = 0$, the source label is continuously one; every job serving occurrence 1 reads this persistent step label, not a distinct source packet. When job j of task τ is first dispatched, it simultaneously snapshots the predecessor label for *every* occurrence h with $\tau^{(h)} = \tau$, or the source label for $h = 1$. A preempted job does not resample. On completion it writes its stored occurrence-specific labels. A non-chain task only consumes CPU. A job takes all its snapshots before it writes any output, so it cannot pass the same source label through two successive chain links in one execution. Under the stipulated first-in, first-out (FIFO) order within each task, each occurrence's output label is nondecreasing over time. Causal failure can nevertheless result from a consumer's premature snapshot or a late sink completion.

Let $T_\gamma(\pi, x)$ be the first time a job at the last chain occurrence writes the value one to its output register. Set $T_\gamma = +\infty$ if no such write occurs. For an integer chain deadline $D \in \mathbb{Z}_{\geq 0}$, define the *causal failure event*

$$e_\pi(x) = \mathbf{1}\{T_\gamma(\pi, x) > D\} \in \{0, 1\}. \tag{2}$$

Evaluating e_π requires simulating only up to boundary D. The event follows a designated source label, not the age of arbitrary packets in a periodic system. Unlike hard job-deadline feasibility, causal success need not be monotone in execution demand; the causal event is therefore compiled without a monotonicity assumption. Figure 1 illustrates the read and write semantics for the nine-job example in Section 4.2.

3.3 Incomplete Dependence Information

We assume that $\Pr(X_i = 1) = 1/2$ for all jobs, *not* that the demands are independent. On selected undirected edges E of a graph $G = (V, E)$, $V = \{0, \ldots, n-1\}$, the model specifies $0 \leq q_{uv} \leq 1$ and requires $\Pr(X_u \neq X_v) \leq q_{uv}$. Equivalently, $g_{uv} = 1 - q_{uv}$ is a lower bound on the agreement probability $\Pr(X_u = X_v)$. Let p_x denote the joint probability of scenario x. The ambiguity polytope is

$$\begin{aligned} \mathcal{P}_G = \Big\{ p \geq 0 : \quad & \sum_x p_x = 1, \qquad \sum_x x_i p_x = \tfrac{1}{2} \quad (i \in V), \\ & \sum_x \mathbf{1}\{x_u \neq x_v\} p_x \leq q_{uv} \quad ((u, v) \in E) \Big\}. \end{aligned} \tag{3}$$

This polytope is nonempty: a half-half mixture of all-zero and all-one vectors satisfies every edge cap. A disagreement upper bound is not a Pearson correlation coefficient, and an absent edge does not imply independence. For fair binary variables, $\mathrm{Cov}(X_u, X_v) \geq (1 - 2q_{uv})/4$, a standard covariance restriction.

For every hard-feasible π, define

$$R(\pi) = \max_{p \in \mathcal{P}_G} \sum_{x \in \{0,1\}^n} p_x e_\pi(x), \qquad R_{\min} = \min_{\pi \in \Pi_{\mathrm{hard}}} R(\pi). \tag{4}$$

The mathematical optimizer returns a globally optimal priority and its certified risk if the finite certification completes, or HardInfeasible if $\Pi_{\mathrm{hard}} = \varnothing$. In particular, with positive execution requirements and $D = 0$, every hard-feasible priority has risk one. The mathematical model permits real caps in $[0, 1]$, including projected caps above $1/2$. The prototype uses integer job times and a finite horizon and certifies results with rational arithmetic. Its input graph presets are a path ($q = 1/5$), a star ($q = 1/4$), and an empty graph; it is not a numerical interface for arbitrary real-valued caps. It reports Exact, HardInfeasible, Unknown for unclosed certification or recognized resource limits, and Error for invalid inputs or unexpected failures. Only Exact returns an optimal risk. The explicit permutation budget can prevent even the hard-feasibility search from completing.

4 Structural Risk and Dependence-Induced Priority Reversal

For a Boolean event f, write $R_G(f) = \max_{p\in\mathcal{P}_G} \mathbb{E}_p[f]$. Its essential support $S(f)$ contains precisely the indices whose individual bit flip can change f. We first identify exact risk expressions, then connect them to a fixed-priority scheduling instance. All graph operations below concern statistical constraints, not task precedence or message links.

4.1 Pairwise and OR Events

Let $d_G(u, v)$ be the shortest-path sum of the edge caps, truncated at one (and equal to one for disconnected vertices). Fairness forces the two diagonal probabilities of (X_u, X_v) to be equal and the two off-diagonal probabilities to be equal. The feasible disagreement range is exactly $[0, d_G(u, v)]$.

Lemma 2 (Exact pairwise risk) *For any Boolean event depending only on X_u, X_v, put $a = (f(0,0) + f(1,1))/2$ and $b = (f(0,1) + f(1,0))/2$. Then*

$$R_G(f) = a + \max(0, b - a)d_G(u, v). \tag{5}$$

Proof For any admissible law the disagreement t of the pair is at most the sum of edge caps along every connecting path, by the union bound. Fairness gives $\mathbb{E}[f] = (1-t)a+tb$. The lower endpoint $t = 0$ is realized by the common-bit law. For the upper endpoint, define $h_i = d_G(u, i)$ and take U uniform on $[0, 1]$ and an independent fair bit Z. Set $X_i = Z \oplus \mathbf{1}\{U < h_i\}$. Then $\Pr(X_i \neq X_j) = |h_i - h_j| \leq q_{ij}$ for every graph edge and $\Pr(X_u \neq X_v) = h_v$. Each bit is fair. Mixing the endpoint laws realizes the entire interval. Maximizing the affine expectation proves the formula. □

For a positive OR event on two bits this reduces to $R_G(X_u \vee X_v) = (1+d_G(u, v))/2$. Thus a tighter statistical path can reduce the worst risk of a priority whose failure is a two-bit OR, independently of other priorities.

Theorem 3 (Tree OR risk) *Let G be a tree, $S \neq \varnothing$, $f = \bigvee_{i\in S} X_i$, and T_S the minimal subtree spanning S. Then*

$$R_G(f) = \frac{1}{2} + \frac{1}{2} \min\left(1, \sum_{e\in E(T_S)} q_e\right). \tag{6}$$

Proof Orient the subtree away from a terminal r. If any terminal is one, either $X_r = 1$ or a subtree edge has a directed transition $0 \to 1$. By fair marginals each transition probability is half the corresponding disagreement probability. The union bound, capped at one, yields the upper bound. Choose masses $p_e \in [0, q_e]$ on the subtree edges with sum $t = \min(1, \sum_e q_e)$. For each such edge, cut the original tree and use the two complementary constant-on-component

assignments, each with mass $p_e/2$. The remaining mass $1-t$ is equally split between the all-zero and all-one assignments. The resulting law has fair marginals. For each subtree edge e, its endpoints disagree precisely in the two assignments for its cut, with total probability p_e. Each selected cut separates terminals, so both complementary assignments trigger the OR. The attained risk is $(1-t)/2+t$. □

Theorem 4 (Uniform connected-support OR risk) *Suppose all graph edges have the same cap q and $G[S]$ is connected, where S is the nonempty support of a positive OR event. With $k=|S|$,*

$$R_G(f) = \frac{1}{2} + \frac{1}{2}\min(1,(k-1)q). \tag{7}$$

Consequently adding any number of edges with the same cap leaves this event's risk unchanged.

Proof For $k=1$, fairness gives $1/2$. Otherwise a spanning tree on $G[S]$ supplies the upper bound as in Theorem 3. Let $w=\min(q,1/(k-1))$. On the terminal bits assign mass $(1-(k-1)w)/2$ to the all-zero state, $(1-w)/2$ to the all-one state and $w/2$ to each distinct singleton-one state. Assign every nonterminal bit the value of a fixed terminal. The law has unit mass, fair marginals, and pairwise disagreement at most w for every pair, including any future edge. Its OR probability is the stated bound. Connectivity is preserved by adding edges. □

The tree bound remains an upper bound for any tree subgraph connecting the terminals in a graph with cycles, but need not be sharp. Section 5.2 gives a counterexample. These formulas exploit marginal fairness and the particular one-sided disagreement constraints; they do not follow from assuming independent demands.

4.2 A Single Statistical Edge Reverses the Optimal Priority

Table 1 specifies a finite job set with chain $A \to B \to C \to B$ and chain deadline $D=17$. All jobs have relative hard deadline 45, and all six strict task-priority orders are hard-feasible (total maximum work is 28 ticks and the latest release is 11). The statistical graph uses the demand indices in the table. These indices do not order the jobs by release time. This instance was discovered by outcome-aware exploration of synthetic task sets, not sampled as an independent prevalence estimate.

Table 1 Nine-job reversal instance; all relative hard deadlines equal 45.

Demand index i	0	1	2	3	4	5	6	7	8
Job	B_1	A_1	C_0	B_{10}	B_{11}	A_{10}	A_0	B_0	C_1
Release	8	11	11	11	11	6	0	2	6
Low demand	3	1	3	2	1	1	3	2	1
High demand	6	3	3	2	2	3	4	2	3

Figure 1 contrasts two demand scenarios under ABC. Increasing only the demand of A_{10} delays the third-stage write beyond the first dispatch of B_1, so the final register remains zero at the causal deadline.

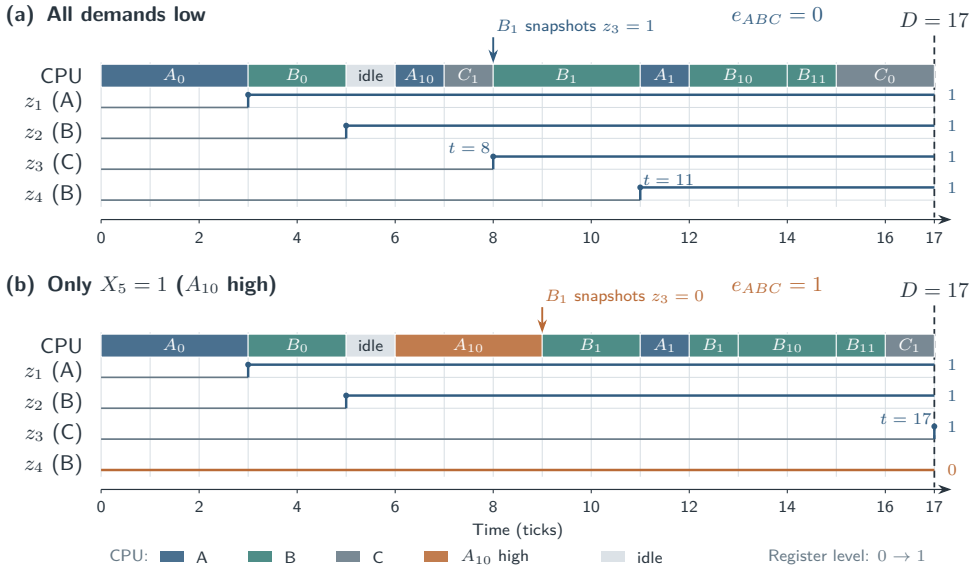

Fig. 1 Single-processor execution and occurrence-specific logical-register traces of the instance in Table 1 under priority ABC. The registers $z_1, \ldots, z_4$ correspond to the four chain occurrences. (a) With all demands low, C_1 writes at boundary 8 before B_1 snapshots z_3, and z_4 becomes one at boundary 11. (b) With only $X_5 = 1$, B_1 snapshots zero at boundary 9; the write at boundary 17 cannot change that snapshot. Register values include writes committed at each boundary. The failure indicator concerns the source label, not a hard job-deadline violation

The deterministic schedule and occurrence-channel rules produce the event identities in Table 2. They were checked on all $6 \cdot 512 = 3072$ priority–demand combinations. The failure events for ABC and CBA have the same number of failing assignments but depend on different pairs of bits: (X_5, X_8) and (X_0, X_8), respectively. Let G_P be the path $0-1-\cdots-8$ and $G_C = G_P \cup \{(0,8)\}$; all edges have the same cap q. No task parameter, hard feasibility condition or event changes between these graphs. The pairwise lemma gives the OR risks directly, since the distances in the path are $3q$ and $8q$ (before truncation), and in the augmented graph are $3q$ and q for the respective pairs.

Table 2 Exact events and risks for every priority. Priority letters are ordered from highest to lowest.

Priority	Failure event	R_{G_P}	R_{G_C}
ABC	$X_5 \vee X_8$	$(1 + \min(1, 3q))/2$	$(1 + \min(1, 3q))/2$
CBA	$X_0 \vee X_8$	$(1 + \min(1, 8q))/2$	$(1 + q)/2$
ACB, CAB	$X_0 \vee \mathbf{1}\{X_1 + X_5 + X_8 \geq 2\}$	$\min(1, \frac{1}{2} + \frac{5}{2}q)$	$\min(1, \frac{1}{2} + q)$
BAC, BCA	1	1	1

Theorem 5 (Single-edge global reversal) *For the instance in Table 1, ABC is the unique robust-optimal priority on G_P for $0 < q < 1/3$, whereas CBA is unique on G_C for $0 < q < 1$. In particular, at every common cap $0 < q < 1/3$ the addition of just $(0, 8)$ reverses the unique global optimum.*

Proof The two-bit and constant risks follow from Lemma 2. Let $f = X_0 \vee \mathbf{1}\{X_1 + X_5 + X_8 \geq 2\}$. On the path, the pointwise inequality

$$f(x) \leq \tfrac{1}{2}(x_0 + x_5) + \tfrac{1}{2} \sum_{i=0}^{4} \mathbf{1}\{x_i \neq x_{i+1}\} \tag{8}$$

holds: when $f = 1$ and both endpoints x_0, x_5 equal zero, the condition forces $x_1 = x_8 = 1$ and the path from 0 through 1 to 5 has at least two transitions; when the endpoints differ it has at least one; when both are one the first term suffices. If $f = 0$ the bound is nonnegative. Taking expectations gives the upper bound $\mathbb{E}[f] \leq 1/2 + 5q/2$. The all-equal fair law attains $1/2$ at $q = 0$; the ten equally weighted assignments in A attain one at $q = 1/5$. Interpolation yields the affine bound at every intermediate q, and the saturated law remains feasible thereafter.

On G_C, if $f = 1$ and $X_0 = 0$, then at least one of X_1 and X_8 equals one. Both latter indices are adjacent to 0. Fairness and the union bound give $R_{G_C}(f) \leq \min(1, 1/2 + q)$. The four equiprobable assignments 011111000, 000001111 and their complements attain one at $q = 1/2$ with all edge disagreements at most $1/2$. Mixing with the all-equal fair law gives equality for all $q \in [0, 1/2]$, and saturation thereafter. These prove all entries of Table 2. Comparing each of the five competitors with ABC on the path and with CBA on the augmented graph proves strict uniqueness on the claimed intervals. □

At $q = 1/4$, the new edge leaves the risk of ABC at $7/8$ but lowers that of CBA from one to $5/8$ (Figure 2). In general, adding an edge shrinks the ambiguity set and cannot increase any fixed policy's risk. It is the unequal reductions of competing policy risks that change the minimizing priority. This demonstrates a scheduling decision not recoverable from hard deadline tests or failure-set sizes alone. It does not imply that adding a statistical constraint typically induces reversal. In particular, a connected-support OR may be insensitive by Theorem 4; the following section generalizes the insensitivity analysis to arbitrary Boolean events.

5 Exact Support-Aware Probability Projection

A variable absent from the essential support of e_π may still constrain the joint law of other variables. The correct reduction is therefore a marginal projection of the ambiguity set, not merely deletion of all unused variables. The reduction preserves the entire set of feasible joint distributions on the retained variables, and hence the maximum probability of any event on those variables.

Theorem 6 (Exact elimination of a nonessential low-degree variable) *Let $S = S(f)$ and $v \notin S$. Suppose v has degree at most two in the current statistical graph. Remove v and its incident edges. If its two neighbors are a, b, insert or tighten their edge to cap*

$$q'_{ab} = \min\{1, q_{av} + q_{vb}, q_{ab}\}, \tag{9}$$

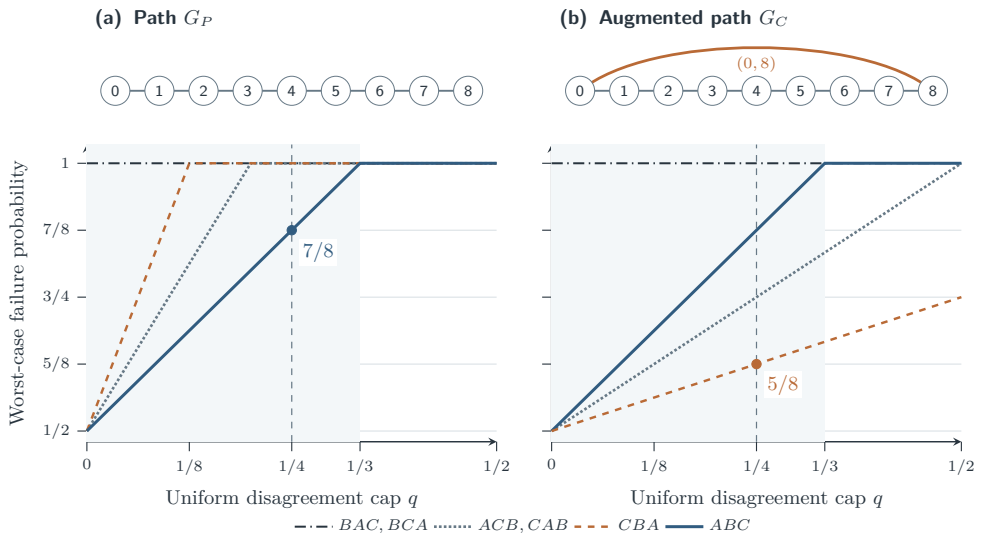

Fig. 2 Exact worst-case failure risks for all six priority orders under (a) path G_P and (b) augmented path G_C. Nodes index job-demand variables, not tasks; the added edge is $(0, 8)$. Shading marks the common strict-reversal interval $0 < q < 1/3$. At the same cap $q = 1/4$, the unique optimum changes from ABC (7/8) to CBA (5/8). Coincident risk functions share a curve; curves are evaluations of Table 2, not fitted simulation data

where a missing ab edge has cap one. For degree zero or one no additional edge is required. Denote the transformed graph by G' and retained variables by $V' = V \setminus \{v\}$. Then

$$\{p|_{V'} : p \in \mathcal{P}_G\} = \mathcal{P}_{G'}, \qquad R_G(f) = R_{G'}(f|_{V'}). \tag{10}$$

Here $f|_{V'}$ is well-defined because v is nonessential, and $p|_{V'}$ denotes a marginal distribution, not a conditional law.

Proof Every original law satisfies the remaining constraints. In the degree-two case the pointwise triangle inequality for disagreement implies $\Pr(X_a \neq X_b) \leq q_{av} + q_{vb}$, and any pre-existing constraint remains valid. Thus every marginal belongs to $\mathcal{P}_{G'}$.

Conversely fix an arbitrary $Q \in \mathcal{P}_{G'}$. For an isolated variable append a fair bit (or copy a retained fair bit); for a leaf append a copy of its unique neighbor. For degree two set $d = Q(X_a \neq X_b)$. If $d = 0$ append $X_v = X_a$. Otherwise let $\theta = \min(1, q_{av}/d)$ and, independently of $X_{V'}$, choose to copy X_b with probability θ or X_a with probability $1 - \theta$. The new bit is fair; its two disagreement probabilities are $\theta d \leq q_{av}$ and $(1 - \theta)d \leq q_{vb}$, the latter by $d \leq q_{av} + q_{vb}$. All retained variables keep their full joint law. This extends every Q to a feasible original law, proving equality of the sets. Nonessentiality preserves the objective under both projection and extension. □

Relation to two-interface components.

A connected component C of variables outside the event's support can also be eliminated if its only external neighbors are a and b. The new cap is the length of the shortest a–b path through C, using edge caps as lengths and truncating at one. Any

existing constraint between the two boundary vertices remains in force. To see sufficiency, fix any retained law with endpoint mismatch d no greater than this budget. Assign hidden vertex w the coordinate $z_w = \min(d, \mathrm{dist}_C(a, w))$, where distances use edges incident to C; set $z_a = 0, z_b = d$. These coordinates differ by at most each incident edge cap. When the endpoints differ, draw a common threshold U uniformly on $[0, d]$, independently of the retained assignment within this case. Set $X_w = X_b$ if $U < z_w$ and $X_w = X_a$ otherwise. When the endpoints agree, copy their common value. The two unequal orientations have equal mass by fairness, so every new bit is fair and each incident edge has mismatch $|z_u - z_v|$. The retained law is unchanged. The case $d = 0$ uses common copies. For $C = \{v\}$ this gives exactly Equation (9); the two-interface rule thus subsumes the degree-two special case. Repeated low-degree elimination remains useful inside a hidden component with more than two boundary vertices, where the whole-component rule does not apply. Neither rule assumes conditional independence of demands in the original ambiguity set.

The local proof is constructive and can be applied recursively. The implementation computes reachable variable labels in a reduced event BDD, removes nonessential vertices of degree at most two in deterministic degree/index order, omits cap-one edges, and records the eliminated vertex and its neighbors with their then-current edge caps. It retains at least one bit to simplify the solver interface. After obtaining an exact risk and a reduced joint law, it replays the elimination trace in reverse, splitting probability atoms with the rational mixing weights in the proof. All original marginal and edge constraints and the original event expectation are checked. The lifting is a construction of a demand law, not a randomization of the scheduling priority. In particular, we neither eliminate jobs from the schedule nor infer independence of hidden variables. Caps induced by addition may exceed $1/2$ and must not be clipped to $1/2$.

Figure 3 shows how to extend a distribution on the retained variables to all demand variables. Under the extended distribution, the failure probability of the certified order equals its worst-case risk, while each other order's failure probability lower-bounds its own worst-case risk.

5.1 Topology Invariance and Safe Exclusion

For each graph obtained by exact elimination, define its signature as the retained vertex set and the matrix of shortest-path caps truncated at one. Every law satisfying the edges satisfies the corresponding shortest-path constraints; conversely the latter imply the original edge constraints. Thus equality of signatures implies equality of the retained ambiguity sets.

Proposition 7 (Risk invariance after projection) *If exact reductions of two statistical graphs for the same event f have the same retained variables and truncated shortest-path cap matrix, then their robust risks agree. Suppose G' adds constraints to G and c has strictly higher risk than an optimal order w under G. If $R_{G'}(e_c) = R_G(e_c)$, then c cannot be optimal under G'.*

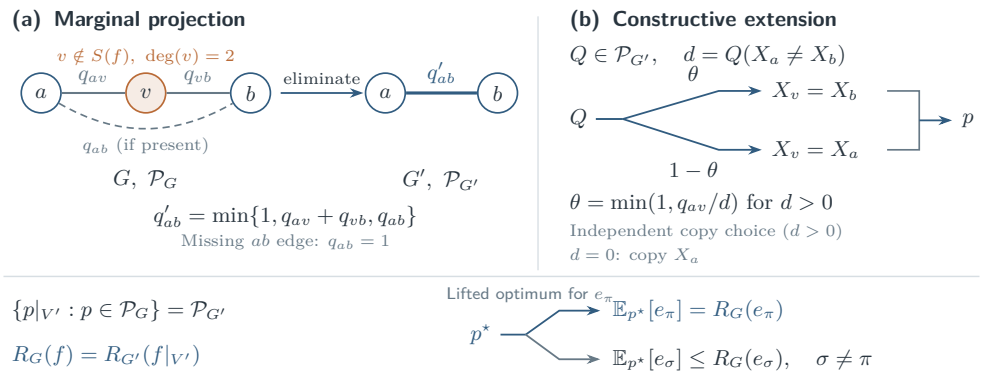

Fig. 3 Exact marginal projection and constructive extension of a nonessential degree-two demand variable. (a) Eliminating v inserts or tightens the retained edge ab. (b) The extension copies one endpoint with a mixing choice independent of the retained assignment for $d > 0$, and copies X_a for $d = 0$. The lifted distribution $p^\star$ attains the worst-case risk for the certified order π. For any other order σ, its failure probability under $p^\star$ is a lower bound on its worst-case risk. The reduction acts on probability constraints, not scheduled jobs

Proof Apply Theorem 6 and the signature equivalence to obtain risk equality. Set inclusion of ambiguity sets gives $R_{G'}(e_w) \leq R_G(e_w) < R_G(e_c) = R_{G'}(e_c)$. Thus c is strictly worse than a feasible policy under the new constraints. □

This is a sufficient, not necessary, condition: different signatures can yield equal risks for a particular event. It applies to non-OR and nonmonotone events, and complements the uniform OR invariance of Theorem 4. The safe exclusion rule uses previously certified old risks; it is not part of the single-graph optimizer timed below.

5.2 Limits of Graph Reduction

The restriction on degree in Theorem 6 is important. Consider three retained fair bits X_0, X_1, X_2 connected to a nonessential center X_3 by caps $1/4$. Let f indicate even parity of the three retained bits. The original risk is $7/8$. If the center is removed and all three retained pair caps are set to $1/2$, the resulting risk is one: assign mass $1/4$ each to $000, 110, 101, 011$. For the original graph, an explicit upper certificate is

$$f(x) \leq 1 - \tfrac{1}{2}(x_0 + x_1 + x_2) + \tfrac{1}{2}x_3 + \tfrac{1}{2}\sum_{i=0}^{2} \mathbf{1}\{x_i \neq x_3\}. \tag{11}$$

The expectation of the right-hand side is at most $7/8$ under every feasible law. A matching rational feasible law is provided in the reproducibility material. Hence even all shortest-path pair bounds can lose higher-order joint constraints after an unjustified elimination.

A separate example shows why the minimum connecting tree is not an exact OR-risk formula for all graphs. Let four terminals $0, 1, 2, 3$ and four auxiliary vertices $4, 5, 6, 7$ be connected by every edge $(i, 4 + j)$ with $i \neq j$, each with cap $1/10$. A minimum Steiner tree connecting all four terminals has five edges: one auxiliary vertex

cannot connect all terminals, whereas two suffice. Equation (6) applied as a tree upper bound gives 3/4. However the full-scenario linear program (LP), with one variable per complete demand assignment, has exact value 29/40, with matching rational primal and dual certificates covering all 256 scenarios. Both examples are Boolean probability counterexamples, not claims of new scheduling instances. They motivate an exact general-purpose fallback when no proved reduction applies.

6 Structure-Aware Exact Priority Optimization

We use a single risk interface: given a hard-feasible deterministic priority order π, return a certified value $R_G(e_\pi)$ and a probability law $Q_\pi \in \mathcal{P}_G$ attaining it. Recovering a maximizing distribution on all original demand variables lets the optimizer evaluate other orders under that distribution, even when their failure events depend on different variables.

6.1 Causal Event Compilation and Shared Scenario Evaluation

An individual event compiler simulates the finite scheduling semantics symbolically through the causal deadline. It branches on unknown demand bits, retains job release and service status, first-dispatch snapshots and occurrence registers, and constructs a reduced ordered BDD of the failure predicate (Bryant 1986). For a complete assignment, the restricted transition sequence is the deterministic schedule. Induction on ticks proves equality of the resulting Boolean event and Equation (2), provided merged states retain all future-relevant information. No independence assumption is used. A successful sink write permits early termination with failure value zero; early termination with failure value one requires a sound impossibility bound. A resource limit must not be interpreted as a Boolean event value.

The outer optimizer also needs evaluations of many priorities at each atom of a probability witness. Instead of compiling a separate BDD for each of them, for a fixed complete demand vector x we compute the success set

$$M(x) = \{\pi \in \Pi_{\text{hard}} : e_\pi(x) = 0\}. \tag{12}$$

This is represented by a bit mask over explicitly enumerated hard-feasible static priorities. At a tick, a shared scheduling state is associated with the mask of priorities reaching it. Release admission is common; the ready FIFO heads are determined by that state. Partition the mask by the highest-priority ready task for each constituent order, execute one transition per distinct selected job and union masks at equal successor states. The transition includes the first-read and completion-write rules. At a successful terminal state, add the orders in its mask to $M(x)$. Any order still unresolved after the writes at boundary D has failed for this assignment. Ready-task choice masks can be cached by the set of ready task identities, while the job identity is taken from the current FIFO head, not cached across different states.

Proposition 8 (Static-policy preservation) *For any complete x, the shared scenario evaluator returns exactly $M(x)$ if it completes without exceeding its resource limit.*

Proof At initialization every feasible policy reaches the common initial state. Suppose a state's mask contains exactly the policies reaching that state at the current tick. The disjoint dispatch partition sends each constituent policy to precisely its deterministic next state. Unioning masks at equal successors preserves membership and does not change any policy's future dispatch rule. When evaluation stops at a successful state, its mask records only orders for which the sink has already written the source label. Induction on ticks establishes the invariant and the result. This is sharing of deterministic policy evaluations, not a scenario-dependent choice of priority. □

The shared evaluator caches $M(x)$ so that different probability laws using the same assignment do not repeat that schedule computation. An individual BDD is compiled only when the exact risk oracle needs it. Equality of canonical BDDs under a fixed variable order allows risk-certificate reuse. The revised implementation verifies event equality by comparing canonical node structures. A matching digest alone does not establish equality. The design differs from the full synchronous baseline, which compiles every priority's complete symbolic event before risk selection.

6.2 Structural Risk Oracle

For completeness, the full-scenario probability LP for an event f has variables p_x and constraints in Equation (3). Its dual can be written

$$\begin{aligned} \min \quad & \alpha_0 + \tfrac{1}{2}\sum_i \alpha_i + \sum_{(u,v)\in E} q_{uv}\beta_{uv} \\ \text{s.t.} \quad & \alpha_0 + \sum_i \alpha_i x_i + \sum_{(u,v)\in E} \beta_{uv}\mathbf{1}\{x_u \neq x_v\} \geq f(x), \quad x \in \{0,1\}^n, \\ & \beta_{uv} \geq 0, \qquad \alpha_0, \alpha_i \in \mathbb{R}. \end{aligned} \tag{13}$$

Assigning mass $1/2$ to each of the all-zero and all-one assignments gives a feasible primal distribution. The risk is bounded by one. A rational feasible primal law and dual upper bound with equal values form an exact certificate. Directly enumerating 2^n assignments provides an independent reference, but is not the main solver.

The structural oracle first checks for a constant event, positive single-bit event, positive OR on a statistical tree, positive OR with uniformly capped connected support, or an unconstrained positive OR. For these cases the constructive formulas of Section 4 (and the trivial constant and unconstrained cases) give an exact value and an attaining law. For the unconstrained OR of at least two bits, a complementary pair of assignments with mixed terminal values achieves one. The implemented closed-form branch is limited to these tests; the more general pairwise lemma is used in structural analysis and is not a separate timed oracle.

Otherwise, apply the exact projection in Theorem 6 and solve a flow LP on the reduced BDD augmented with the values of earlier bits still needed to check an edge constraint. For an ordering $1, \ldots, n'$ of the retained bits, define layer p immediately before assigning bit p. Let $H_p = \{i < p : \exists j \geq p,\ (i,j) \in E'\}$, where E' is the reduced graph edge set. A layer-p state stores the residual BDD node and the values of H_p. An arc chooses the next bit, advances the BDD and updates the history. The LP imposes

unit source flow, flow conservation, marginal probability constraints and disagreement constraints. Each edge disagreement is counted when its later endpoint is assigned. The LP maximizes the mass reaching the failure terminal (BDD value one).

Proposition 9 (Exact frontier-flow risk) *The optimum of this LP is the reduced event's robust risk. With B_p the set of possible residual BDD nodes at layer p and $w_p = |H_p|$, the number of arcs is at most $2\sum_{p=1}^{n'} |B_p| 2^{w_p}$.*

Proof Push each complete demand atom through its unique bit-labeled path. The path contributes the atom's mass to the marginal and disagreement constraints and to the failure objective with exactly the same coefficients as in the original probability LP. Conversely, every feasible unit flow in the layered acyclic graph decomposes into complete paths. Their bit labels and masses yield a feasible joint law with the same event expectation. Enumerating residual nodes, history bit vectors and two possible outgoing bits gives the bound. □

The implementation reconstructs rational primal and dual solutions to the flow LP and checks feasibility and equality of their objective values. It stores the recovered laws and master records; the independent original-space audit additionally exports rational atom-space primal and dual certificates for the tested events. The reduced law is lifted along the recorded elimination trace. Its exact mass, original marginal probabilities, all original edge caps and original BDD expectation are verified before it becomes a cross-priority witness. Projection can decrease the number of bits but can also introduce fill edges and increase the number of atoms during lifting. Consequently neither BDD size nor dimension alone guarantees an end-to-end saving.

6.3 Finite-Witness Master and Global Certificate

Let $W \subseteq \mathcal{P}_G$ be a finite nonempty collection of certified laws. Define

$$\ell_W(\pi) = \max_{Q \in W} \mathbb{E}_Q[e_\pi], \qquad L(W) = \min_{\pi \in \Pi_{\text{hard}}} \ell_W(\pi). \tag{14}$$

Every $\ell_W(\pi)$ is a lower bound on the robust risk of that fixed priority, so $L(W) \le R_{\min}$. Each expectation can be computed by summing the witness's atom masses for which $\pi \notin M(x)$. Incremental evaluation adds only new witness rows; all use the common original graph and full-dimensional demand vectors. A candidate's certified true risk gives an upper bound on the global optimum. The two bounds are tightened in one exact loop (Algorithm 1). Among tied candidates, the implementation maximizes the minimum Kendall distance to previously certified orders, using a fixed order to break any remaining tie. Kendall distance counts pairs of tasks ranked in opposite orders. It affects candidate selection cost, not correctness.

Theorem 10 (Exactness and finite termination) *Suppose hard-feasibility screening completes and $\Pi_{\text{hard}} \neq \varnothing$. If the oracle and shared evaluator terminate with exact results, Algorithm 1 terminates with one globally optimal hard-feasible priority and its exact robust risk after at most $|\Pi_{\text{hard}}| + 1$ master selections.*

Algorithm 1. Structure-aware finite-witness priority optimization.

Require: Finite job set, chain and deadline, statistical graph G
Ensure: Certified optimal static priority and risk, or no exact result

Enumerate hard-feasible static priorities Π_{hard}
if hard screening cannot finish **then**
    return UNKNOWN
end if
if $\Pi_{\mathrm{hard}} = \varnothing$ **then**
    return HARDINFEASIBLE
end if
For $D = 0$, use the constant-one failure event
$W \leftarrow \{Q_{\mathrm{common}}\}$; $U \leftarrow +\infty$; initialize caches
loop
    Update witness expectations using cached shared masks $M(x)$
    $L \leftarrow \min_{\pi} \ell_W(\pi)$; choose minimizing c with fixed tie rule
    Compile e_c if needed; obtain (r, Q) from the exact structural oracle
    If certification or compilation fails, return no exact result
    If $r < U$, set $(U, \pi^\star) \leftarrow (r, c)$
    if $L = U$ **then**
        return $(\pi^\star, U, L, W)$
    end if
    $W \leftarrow W \cup \{Q\}$ ▷ full-dimensional, validated law
end loop

Proof All witnesses, including the initial common-bit law, belong to $\mathcal{P}_G$. Consequently $L \leq R_{\min} \leq U$ after each exact incumbent update. At equality the returned priority attains U and is globally optimal. Suppose $L < U$ after evaluating the candidate c. If its attained risk were equal to L, the incumbent would be at most L; therefore its attained risk exceeds $L = \ell_W(c)$. The maximizing law returned by the oracle cannot already be in W, and its addition makes $\ell_W(c)$ equal to $R_G(e_c)$. For any subsequent master selection before closure, every previously certified candidate has witness risk equal to its robust risk, hence at least the current incumbent upper bound. None can be a minimizer with lower bound strictly below that upper bound. Thus each nonclosing iteration certifies a previously uncertified candidate. Finiteness establishes the claimed bound, allowing a final equality check. Exact status requires a closed certificate; failure to compute a certificate does not imply infeasibility. □

A single law may not suffice for global certification. For finite strategies and the compact convex ambiguity set, the largest possible one-law bound is $V_{\mathrm{mix}} = \max_Q \min_\pi \mathbb{E}_Q[e_\pi]$, which equals the minimax value allowing randomized priorities by the finite-dimensional minimax theorem. A single law certifies the deterministic optimum iff $V_{\mathrm{mix}} = R_{\min}$. The master in Equation (14) retains static priorities; it does not solve that randomized relaxation. A gives the concrete gap in the reversal model and a two-law certificate. At that parameter point, more than one feasible joint distribution is necessary to certify the deterministic optimum, and two suffice.

Computational Scope.

The explicit master can contain up to $m!$ orders for m task types. With K distinct witness assignments, the cached shared scenario evaluator stores up to K success masks of length $|\Pi_{\text{hard}}|$ plus schedule states and transition caches. A witness can require several atoms; worst-case scheduling exploration and flow states remain exponential. Structural checks and projection reduce individual probability problems when their supports permit it; no polynomial overall complexity or universal advantage over full symbolic compilation is claimed.

AI-assisted research support.

ChatGPT and Codex (OpenAI; exact model versions were not consistently recorded) were used as research-support tools during literature research, methodological development, mathematical derivation and verification, computational implementation, and experimental analysis. The authors reviewed and verified the AI-assisted derivations, code, and analyses and take responsibility for the research methods and results.

7 Evaluation

We organize the evaluation around semantic and certificate correctness, dependence-induced decision changes, cost of structural certification, and scale limits. The original frozen optimizer uses the structure-aware oracle and finite-witness master described above. We distinguish mathematical checks, development-set experiments, a prospectively fixed synthetic benchmark, and a subsequent four-arm ablation using the same benchmark inputs and the revised interface. Correctness is checked against a separately coded reference implementation; the prospective benchmark uses previously unseen synthetic inputs. The timing measurements are internal experiments, rather than an external replication.

7.1 Correctness and Decision Structure

The original single-edge example has all six priority events checked on 3072 demand–priority combinations; Table 2 and A give the exact risks and constructive proofs. The strict reversal is not restricted to two-bit OR competition: two outcome-selected variants of a different nine-job parent model with chain $A \to B \to A \to B$ exhibit strict reversal from BAH to ABH when edge $(3, 8)$ is added. Exact piecewise risk certificates give reversal intervals $(0, 1/3)$ and $(0, 1/2)$ respectively; in the second variant both competitors have more than two essential variables. Because these variants share one parent and were selected after exploration, they establish further existence examples but are not independent confirmation of prevalence. The accompanying Supplementary Materials specify their job parameters and risk functions.

The structural projection theorem was tested using 300 Boolean-event instances with 3–7 bits, covering connected trees, cycles, disconnected graphs and heterogeneous caps. A fixed seed selected inputs before testing. Full and reduced atom LPs yielded 600 matching rational primal–dual solutions; 218 instances reduced dimension. The sum of

original scenario counts was 14,880 versus 5214 after reduction. These are mathematical verification instances rather than scheduling-generated tasks. The independent graph counterexamples in Section 5.2 are also checked with rational certificates.

In a separate retrospective topology-change analysis of 144 previously frozen graph/parameter comparisons, the projected-graph signature certified invariance for 2662 of 3456 policy conditions, including 502 nonconstant, non-OR events. Among 966 orders strictly worse than the previous unique optimum, it safely excluded 676. The earlier connected-OR criterion excluded 234, of which 174 overlapped. These counts measure safe exclusion on that historical suite; the topology-change procedure is not included in the main single-graph timing gains. Across the three new task-generation cohorts used earlier in the development phase (288 plus 96 graph comparisons), no strict unique-to-unique reversal was observed. These comparisons establish the existence of strict reversals in the constructed instances and verify safe exclusion on the historical suite; their frequency in other workloads remains unknown.

7.2 Comparison Methods and Experimental Protocol

All timed arms solve the identical static-priority optimization problem and use the same deterministic semantics and probability model. The baselines and ablations are:

1. **SMPSC+W**: the earlier synchronous multi-priority symbolic compiler, event equality/dominance, exact BDD-flow and probability-witness transfer. This internal baseline solves the same optimization problem as the proposed method.
2. **Shared family**: shared scenario evaluation with partial-order family search and common-failure witness lower bounds; exact BDD-flow for candidates.
3. **Active witness**: the finite-witness master and shared scenario evaluator, using unprojected exact BDD-flow.
4. **Projection only**: the same master, using the exact graph projection with flow, without the closed-form risk branch.
5. **Structural**: Algorithm 1 with projection and the specified closed-form branch.

The development suite consists of 21 earlier synthetic models and 48 graph conditions, including six complete six-stage chain stress conditions. Four arms (excluding Shared family) were tested three times per condition in randomized order, producing 576 full runs. This suite contributed to method development and is reported as exploratory ablation, not independent confirmation. The later benchmark freezes 12 new job sets before measurement: for each $m = 3, 4, 5, 6$ one pipeline visiting all task types, one two-stage chain with background tasks, and one four-occurrence reentrant chain $A \to B \to A \to C$. Releases and demands are generated by fixed seeds, with 4–9 jobs and relative hard deadlines of 100. Each instance uses a path with cap 1/5 and a star centered at index zero with cap 1/4. The regimes differ in both topology and cap, unlike the controlled single-edge theorem. No model is removed after observing its risk. The input and frozen algorithm hashes precede the experiment. All strict priorities in the new benchmark are hard-feasible. Separate boundary tests for the

revised entry cover a mixed hard-feasible domain, an empty domain, zero causal deadline, tied optima, invalid inputs and forced resource limits; these are not performance workloads.

The five methods run three times per condition in randomized arm order within each repeat. Wall time includes hard-feasibility handling, event compilation, probability certification and priority search, with the frozen solver resource limits retained. For each method and condition we use the median of three timings. We report the geometric mean of paired median ratios, counting a win if the structural median is strictly lower. Timing repetitions are not independent workloads. Separate measurements in fresh processes record peak resident set size (RSS) for representative cases. The implementation is an offline Python/SciPy prototype; saved measurements are machine-specific. The frozen records retain exceptions and unsuccessful runs rather than assigning them a risk or speedup.

7.3 Independent Checks of Optimal Risk on the New Benchmark

A separately coded tick simulator evaluates all priority–demand combinations in the 12 models (906,336 evaluations). The simulator identifies 79 distinct events across the models, counting within each model. Independently checking pointwise dominance leaves 15 nondominated events in total. The independent original-space LP then certifies the minimum risk and checks the actual selected candidates. In total, 36 original-space LP solves and 5504 exact rational dual inequalities were checked. All 348 exact main-experiment results agreed with this reference. The 72 structural runs were independently audited, including 114 master iterations, 108 oracle records, 126 full-dimensional witnesses, and 822 distinct event–law expectations.

The original SMPSC+W configuration completed 60 of its 72 runs; it reached its synchronous policy-transition budget of $4{\cdot}10^5$ on four conditions (two six-task nine-job models, two graphs per model). Every other arm completed all 72 runs. The missing 12 baseline runs were repeated after increasing the state-transition budget to 10^7; all 12 completed and agreed with the reference. This is a follow-up budget sensitivity check using the direct synchronous-compiler entry followed by the same dominance and exact risk backend, not part of the original randomized timing batch. Only these four condition medians are replaced in the extended follow-up row. No speedup is calculated using a failed baseline run.

7.4 End-to-End Cost and Ablation

Table 3 reports the developmental ablation. Across 48 conditions, all 576 runs give the same optimum within each condition. The structural method reduces the geometric mean time ratio from 0.829 (Active witness) to 0.765 relative to SMPSC+W. Its own time ratio relative to Active witness is 0.923. Of 210 candidate certifications in 144 structural runs, 63 use closed forms and 147 use flow; both the unprojected and projection-only active-witness arms use 210 flow calls. Among the closed forms, 27

are constant events, 27 tree OR events and 9 single-bit events. In all six full six-stage conditions, no event structure enables a closed form or variable elimination; the structural method is slower than the strong baseline on every one of these conditions.

Table 3 Development-suite ablation (48 conditions). Time ratios relative to SMPSC+W; lower is better.

Method	All conditions	42 regular	6 full-stage stress
SMPSC+W	1.000	1.000	1.000
Active witness	0.829	0.757	1.572
Projection only	0.811	0.738	1.561
Structural	0.765	0.690	1.586

Table 4 gives the main prospective synthetic comparison. The structural method is faster on 14 of 20 conditions solved by both methods under the original budget, with geometric mean ratio 0.410. The gain across the 20 jointly solved conditions partly reflects the zero-risk cases and the cost of compiling unnecessary full events. On the six zero-risk conditions solved by both methods within the original transition budget, the geometric mean time ratio was 0.105. Ten of the 24 conditions have zero optimal risk, and all four baseline-budget failures occur in that group. For the 14 nonzero-risk conditions, the ratio relative to SMPSC+W is 0.736 (8 wins), and the comparisons with the other stronger arms are closer to one. Using the separately retimed extended-budget results for the four missing conditions gives 0.268 over all 24 conditions, but that figure is a sensitivity analysis rather than a same-batch estimate. Figure 4(a) shows the individual ratios, making variation and the extended-budget cases visible.

Table 4 New frozen synthetic benchmark. Geometric mean of paired structural/comparator median time ratios; wins are strict. The extended-cap line is a separate sensitivity summary.

	Available conditions		Nonzero risk	
Comparator	Ratio	Wins	Ratio	Wins
SMPSC+W (original cap)	0.410	14/20	0.736	8/14
SMPSC+W (extended follow-up)	0.268	18/24	0.736	8/14
Shared family	0.831	17/24	0.953	8/14
Active witness	0.832	19/24	0.919	9/14
Projection only	0.860	16/24	0.939	7/14

All three active-witness arms perform 108 complete candidate certifications over 72 runs. The unprojected and projection-only arms each use 108 flow solves; the structural arm uses 66 flow solves and 42 closed forms (30 constant, 12 single-bit). It eliminates 87 bits in aggregate, whereas the projection-only arm eliminates 339 bits because it also

processes events handled directly by the closed forms in the structural arm. No multi-bit OR closed form is used on this new cohort. Relative to Active witness, Projection only has a geometric mean runtime ratio of 0.968 overall and 0.978 on conditions with nonzero optimal risk. These observations isolate the modest projection-only benefit in this timing batch; the original three-arm ablation does not identify an interaction between the two mechanisms. On nonzero-risk conditions, Structural is only modestly faster than Projection only in the geometric mean. Against SMPSC+W, its time ratio on the eight full-pipeline conditions is 1.091, with four wins. The longer chains and the weak or absent support reduction expose costs of the explicit witness master.

7.5 Matched Two-Factor Ablation

To identify which risk-oracle mechanisms account for the earlier time differences, we reran the same 24 frozen conditions with four arms in one versioned implementation. Only the projection and closed-form switches differ. Five repetitions per condition were executed in randomized arm order within each repetition (480 runs); the inputs were already known from the prospective benchmark, so this is a follow-up ablation, not a new independent workload cohort. All arms use the same canonical event cache, shared evaluation, hard screening, full-dimensional law checks and stopping rule. The timing environment was a Linux x86-64 cloud VM with five visible logical CPUs (AMD EPYC 9V74), Python 3.13.5, NumPy 2.3.5 and SciPy 1.17.0; OpenBLAS and OMP thread counts were set to one. The VM was not CPU-pinned. We compare paired condition medians of five measurements, not pooled single-run ratios. This batch is not combined with the earlier baseline timing batch.

Table 5 Matched factorial ablation. Geometric mean of paired median ratios relative to the arm with neither risk-oracle mechanism; wins are strict.

	All 24 conditions		14 nonzero risks	
Configuration	Ratio	Wins	Ratio	Wins
Neither	1.000	–	1.000	–
Projection only	0.966	15/24	0.987	7/14
Closed forms only	0.839	16/24	0.933	7/14
Both (Structural)	0.819	18/24	0.924	8/14

All 480 runs match the original-space independent optimum. The separate audit regenerates the 906,336 deterministic priority–demand evaluations and checks 905 original-space witness laws, 760 master steps and 720 risk-oracle records. The 36 atom-space LP certificates, covering 5504 rational dual inequalities, are also exported. The reference uses the common hard-screen implementation, which is separately tested on boundary inputs. In each arm with closed forms enabled, 70 of the 180 candidate certifications across 120 runs used a closed form in place of a flow solve. Of these 70 events, 50 were constant and 20 depended on a single bit. No multi-bit OR closed form is called. The projection-only arm removes 565 bits across 180 certifications; with

both mechanisms enabled it removes 145 bits across the remaining 110 flow solves. The removal counts include repeats and are not independent task models.

Figure 4(b) contrasts the four arms for the full cohort and its nonzero-risk subset. The geometric mean runtime ratio for Both relative to Closed forms only is 0.976 overall and 0.990 on conditions with nonzero optimal risk. Define the multiplicative interaction as $T_{11}T_{00}/(T_{10}T_{01})$ per condition, where the two subscripts indicate projection and closed forms; its geometric mean is 1.011 overall and 1.004 for nonzero risks. These descriptive results do not support a synergistic speedup. They support a common exact interface and show that the simple closed-only arm is a competitive alternative on these workloads. We retain projection as an optional exact reduction for events with eliminable hidden variables, not as a prerequisite for correctness or a guarantee of material speedup. All four arms are released to make that choice reproducible.

The revised entry also passes 29 interface tests (including all four configurations for zero deadline, empty or mixed hard-feasible sets, tied optima, invalid inputs, and forced enumeration or scenario-budget failures). A separate tick-level reference checks 4224 hard-deadline demand–priority cases across tightened relative deadlines; four injected uncertified-flow outcomes also correctly return no optimum. The exact status is reserved for a closed certificate; unsuccessful certification is not reported as infeasibility.

7.6 Resource Use and Validity Limits

Twenty fresh-process RSS probes cover four representative path models and five arms; 18 complete and two baseline probes hit the default transition cap. Successful peak RSS is approximately 133.1–136.5 MiB including Python and numerical-library overhead. The six-task pipeline consumes approximately 136.5 MiB for Structural versus 134.5 MiB for SMPSC+W. The evidence does not demonstrate memory savings. The largest prospective workload has six task types (720 priority orders) and nine jobs. The benchmark does not establish scalability beyond this range.

The synthetic task sets and loose hard deadlines isolate the causal-risk decision, but do not characterize an operational workload distribution. Fair binary demands and one-sided disagreement caps are restrictive; the projection proofs and formulas do not automatically extend to arbitrary marginals, multilevel demands or richer moment constraints. In the benchmark, demand indices determine which jobs are neighbors in the path and star presets. The experiments use specified constraints throughout and draw on synthetic inputs alone. Additional edges may be incorrect if the underlying statistical evidence is unreliable. Real-system traces, timing overheads of an operating-system implementation, periodic infinite-horizon behavior and multicore scheduling are not evaluated. Absolute timing measurements are exploratory, with limited repetition and without a cross-machine replication. The prospective benchmark supports exactness and conditional efficiency of the frozen candidate; the follow-up ablation supports a modest conditional net contribution of projection on these inputs. Neither establishes broad performance, interaction significance or reversal prevalence.

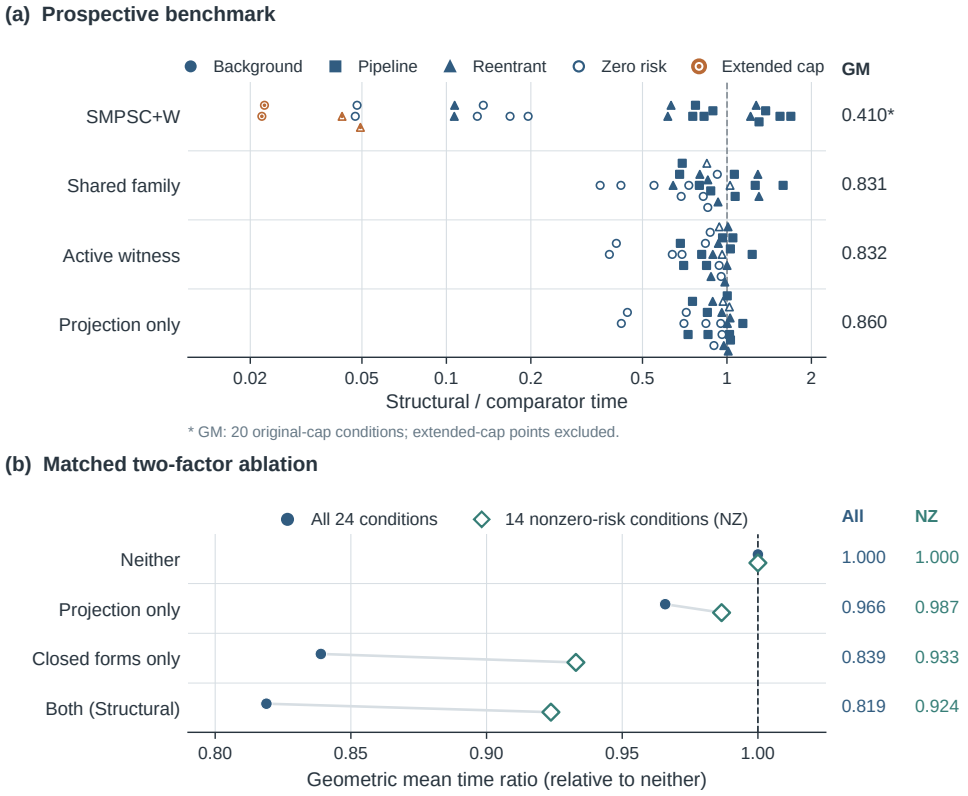

Fig. 4 End-to-end timing comparisons from separate experimental batches on the same 24 frozen conditions. (a) Structural/comparator ratios of condition medians (three repetitions); marker shape denotes workload structure, hollow markers denote zero risk, and orange outlines mark SMPSC+W (synchronous multi-priority symbolic compilation with witness transfer) conditions retimed separately with an increased synchronous-compiler state-transition budget. Vertical offsets reduce overlap only. GM denotes the geometric mean over jointly solved original-transition-budget conditions (20 for SMPSC+W; 24 otherwise). (b) Matched two-factor ablation (five repetitions): geometric means of paired condition-median ratios relative to neither mechanism, for all 24 and 14 nonzero-risk conditions. Connecting segments compare subsets, not uncertainty intervals. Dashed vertical lines mark parity; lower is better

8 Discussion and Conclusions

Static priority scheduling produces distinct Boolean events; the statistical graph changes their allowable probability mass. A single extra constraint can therefore change the global optimum even though all priority orders in the example satisfy the hard job deadlines. Conversely, structural invariance can prove that certain candidates cannot benefit from a graph change. The event support is an appropriate interface only if the constraints induced by hidden variables are preserved. For any Boolean event, exact low-degree elimination outside its support preserves the full feasible set of joint distributions on the retained variables. The degree-three example shows that pairwise elimination can lose joint constraints; the Steiner example shows that a connecting-tree

upper bound on OR risk need not be tight. The local rule specializes existing marginal-extension reasoning; the contribution here is the connection between such structural reductions, scheduling-generated events and full-space optimization witnesses.

The resulting optimizer does not require a separate search strategy for every event type. Both the closed-form and flow branches return the exact risk and a feasible joint distribution on all original demand variables that attains that risk. This enables global deterministic-priority certification with a finite witness set. The evaluation confirms exact agreement with independent references on the tested finite workloads. The matched ablation identifies the simpler Closed forms only configuration as competitive, with most observed gains attributable to constant and single-bit events. Witness evaluation can dominate when events offer no structural reduction. Future work could derive statistically justified ambiguity constraints from real workloads, extend the analysis to broader demand distributions, and represent static priority orders without enumerating all permutations.

Declarations

Author Contributions.

H.J.C. contributed to the conceptualization and methodology of the study and wrote the original manuscript. H.C. made substantial contributions to the investigation, data curation, and visualization. W.H. and Z.L. contributed to software development and implementation. All authors contributed to the review and editing of the manuscript and approved the final version.

Funding.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Competing Interests.

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability.

The synthetic workloads, experimental records, and verification certificates supporting this study are provided in the accompanying supplementary information and research artifact.

Ethics Approval and Consent to Participate.

Not applicable. This study used synthetic task sets and computational experiments and did not involve human participants, identifiable personal data, animals, or plants.

A Constructive Certificates for the Reversal Instance

All bit strings below are in the order $X_0, \ldots, X_8$. For the four-bit competitor's path risk, the endpoint distribution at $q = 1/5$ assigns equal probability $1/10$ to

```
000001111  000011111  000111111  001111011  011111110
100000000  110000001  111000100  111100000  111110000.
```

Each bit has five ones among the ten assignments. Each of the eight adjacent pairs differs in at most two of the ten assignments. The four-bit failure indicator equals one on all ten assignments. Together with Equation (8), this certifies the continuous path risk function. The four equiprobable strings in the proof of Theorem 5 similarly certify the augmented graph risk. The supplementary executable verifier checks the event identities and all endpoint constraints using exact fractions, without an LP solver.

The deterministic optimum at path cap $q = 1/4$ is $7/8$. The independently certified randomized relaxation has value $13/16$ (a strict gap of $1/16$). Two attaining laws Q_{ABC} and Q_{ACB} have expectation rows over the distinct event classes $(ABC, ACB/CAB, CBA, BAC/BCA)$ equal to $(7/8, 1/2, 7/8, 1)$ and $(5/8, 1, 7/8, 1)$, respectively. Their coordinatewise maxima have minimum $7/8$, matching the certified upper bound of ABC. The master thus closes with an original-space two-law certificate, and the strict randomized gap proves that two laws are necessary at this parameter point. The full rational primal and dual records and the game verification script are retained in the frozen evidence package; this result concerns the selected instance, not a bound on the number of laws for general workloads.

Supplementary Information

Online Resource 1. Supplementary Information containing two additional strict-reversal variants, exact probability-graph counterexample certificates, and condition-level timing data.
Online Resource 2. Research Artifact containing source code, synthetic inputs, exact certificates, raw experimental records, and executable verification scripts.

References

M. T. A. Ahmad, J. Pestana, and M. Baunach, "Packet-based data latency and schedulability optimization in directed acyclic graphs of multirate automotive cause–effect chains," *Journal of Systems Architecture*, vol. 179, art. 103909, 2026. https://doi.org/10.1016/j.sysarc.2026.103909.

N. C. Audsley, "On Priority Assignment in Fixed Priority Scheduling," *Information Processing Letters*, vol. 79, no. 1, pp. 39–44, 2001. https://doi.org/10.1016/S0020-0190(00)00165-4.

F. Barahona and A. R. Mahjoub, "On the cut polytope," *Mathematical Programming*, vol. 36, pp. 157–173, 1986. https://doi.org/10.1007/BF02592023.

S. Ben-Amor, L. Cucu-Grosjean, M. Mezouak, and Y. Sorel, "Probabilistic Schedulability Analysis for Precedence Constrained Tasks on Partitioned Multi-core," *ETFA*, pp. 345–352, 2020. https://doi.org/10.1109/ETFA46521.2020.9211973.

M. Bodur, T. C. Y. Chan, and I. Y. Zhu, "Network Flow Models for Robust Binary Optimization with Selective Adaptability," *INFORMS Journal on Computing*, 2026. https://doi.org/10.1287/ijoc.2024.0718.

M. Branda, "Distributionally robust fixed interval scheduling on parallel identical machines under uncertain finishing times," *Computers & Operations Research*, vol. 98, pp. 231–239, 2018. https://doi.org/10.1016/j.cor.2018.05.025.

R. E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation," *IEEE Trans. Comput.*, vol. 35, no. 8, pp. 677–691, 1986. https://doi.org/10.1109/TC.1986.1676819.

L. Chen, W. Ma, K. Natarajan, D. Simchi-Levi, and Z. Yan, "Distributionally Robust Linear and Discrete Optimization with Marginals," *Operations Research*, vol. 70, no. 3, pp. 1822–1834, 2022. https://doi.org/10.1287/opre.2021.2243.

H. Choi, M. Karimi, and H. Kim, "Chain-based Fixed-Priority Scheduling of Loosely-Dependent Tasks," *ICCD*, pp. 631–639, 2020. https://doi.org/10.1109/ICCD50377.2020.00109.

A. Classen, P. Heymans, P.-Y. Schobbens, A. Legay, and J.-F. Raskin, "Model Checking Lots of Systems: Efficient Verification of Temporal Properties in Software Product Lines," *ICSE*, pp. 335–344, 2010. https://doi.org/10.1145/1806799.1806850.

M. Dürr, G. von der Brüggen, K.-H. Chen, and J.-J. Chen, "End-to-End Timing Analysis of Sporadic Cause-Effect Chains in Distributed Systems," *ACM Trans. Embedded Comput. Syst.*, vol. 18, no. 5s, art. 58, 2019. https://doi.org/10.1145/3358181.

A. Friebe, F. Marković, A. V. Papadopoulos, and T. Nolte, "Efficiently Bounding Deadline Miss Probabilities of Markov Chain Real-Time Tasks," *Real-Time Systems*, vol. 60, pp. 443–490, 2024. https://doi.org/10.1007/s11241-024-09431-7.

M. Günzel, H. Teper, K.-H. Chen, G. von der Brüggen, and J.-J. Chen, "On the Equivalence of Maximum Reaction Time and Maximum Data Age for Cause-Effect Chains," in *Proc. ECRTS*, LIPIcs, vol. 262, art. 10, 2023. https://doi.org/10.4230/LIPIcs.ECRTS.2023.10.

M. Günzel, N. Ueter, K.-H. Chen, G. von der Brüggen, and J.-J. Chen, "Probabilistic Reaction Time Analysis," *ACM Transactions on Embedded Computing Systems*, vol. 22, no. 5s, art. 143, 2023. https://doi.org/10.1145/3609390.

M. Günzel, H. Teper, G. von der Brüggen, and J.-J. Chen, "End-To-End Latency of Cause-Effect Chains: A Tutorial," *ACM Trans. Embedded Comput. Syst.*, vol. 24, no. 1, art. 22, 2025. https://doi.org/10.1145/3703630.

R. Ha and J. W.-S. Liu, "Validating Timing Constraints in Multiprocessor and Distributed Real-Time Systems," *ICDCS*, pp. 162–171, 1994. https://doi.org/10.1109/ICDCS.1994.302407.

T. Hailperin, "Best Possible Inequalities for the Probability of a Logical Function of Events," *American Mathematical Monthly*, vol. 72, no. 4, pp. 343–359, 1965. https://doi.org/10.1080/00029890.1965.11970533.

T. Han and K. Kim, "Minimizing Probabilistic End-to-end Latencies of Autonomous Driving Systems," *RTAS*, pp. 27–39, 2023. https://doi.org/10.1109/RTAS58335.2023.00010.

M. Huber and N. Marić, "Admissible Bernoulli correlations," *Journal of Statistical Distributions and Applications*, vol. 6, art. 2, 2019. https://doi.org/10.1186/s40488-019-0091-5.

D. Kuhn, S. Shafiee, and W. Wiesemann, "Distributionally Robust Optimization," *Acta Numerica*, vol. 34, pp. 579–804, 2025. https://doi.org/10.1017/S0962492924000084.

H. Lee, Y. Choi, T. Han, and K. Kim, "Probabilistically Guaranteeing End-to-End Latencies in Autonomous Vehicle Computing Systems," *IEEE Trans. Comput.*, vol. 71, no. 12, pp. 3361–3374, 2022. https://doi.org/10.1109/TC.2022.3152105.

C. L. Liu and J. W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment," *J. ACM*, vol. 20, no. 1, pp. 46–61, 1973. https://doi.org/10.1145/321738.321743.

F. Marković, P. Roux, S. Bozhko, A. V. Papadopoulos, and B. B. Brandenburg, "CTA: A Correlation-Tolerant Analysis of the Deadline-Failure Probability of Dependent Tasks," *RTSS*, pp. 317–330, 2023. https://doi.org/10.1109/RTSS59052.2023.00035.

F. Marković, G. von der Brüggen, M. Günzel, J.-J. Chen, and B. B. Brandenburg, "A Distribution-Agnostic and Correlation-Aware Analysis of Periodic Tasks," *RTSS*, pp. 215–228, 2024. https://doi.org/10.1109/RTSS62706.2024.00027.

D. Maxim, O. Buffet, L. Santinelli, L. Cucu-Grosjean, and R. I. Davis, "Optimal Priority Assignment Algorithms for Probabilistic Real-Time Systems," *RTNS*, pp. 129–138, 2011.

D. Padmanabhan and K. Natarajan, "Tree Bounds for Sums of Bernoulli Random Variables: A Linear Optimization Approach," *INFORMS Journal on Optimization*, vol. 3, no. 1, pp. 23–45, 2021. https://doi.org/10.1287/ijoo.2019.0038.

F. Paladino, A. Biondi, E. Bini, and P. Pazzaglia, "Optimizing Per-Core Priorities to Minimize End-To-End Latencies," *ECRTS*, LIPIcs, vol. 298, art. 6, 2024. https://doi.org/10.4230/LIPIcs.ECRTS.2024.6.

J. Ren, S. Li, R. Bi, J. Zhou, Y. Zheng, X. Qi, and X. He, "Efficient maximum reaction time analysis for data chains of real-time tasks in multiprocessor systems," *Journal of Systems Architecture*, vol. 168, art. 103524, 2025. https://doi.org/10.1016/j.sysarc.2025.103524.

Y. Tang, X. Jiang, N. Guan, D. Ji, X. Luo, and W. Yi, "Comparing Communication Paradigms in Cause-Effect Chains," *IEEE Trans. Comput.*, vol. 72, no. 1, pp. 82–96, 2023. https://doi.org/10.1109/TC.2022.3197082.

N. N. Vorob'ev, "Consistent Families of Measures and Their Extensions," *Theory of Probability and Its Applications*, vol. 7, no. 2, pp. 147–163, 1962. https://doi.org/10.1137/1107014.

Z. Wang, F. Li, Q. Deng, T. Zhang, and Y. Ma, "Ensuring schedulability and end-to-end latency in DAG-based tasks for AUTOSAR," *Journal of Systems Architecture*, vol. 178, art. 103893, 2026. https://doi.org/10.1016/j.sysarc.2026.103893.

B. Zeng and L. Zhao, "Solving two-stage robust optimization problems using a column-and-constraint generation method," *Operations Research Letters*, vol. 41, no. 5, pp. 457–461, 2013. https://doi.org/10.1016/j.orl.2013.05.003.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- OnlineResource1SupplementaryInformation.pdf
- OnlineResource2ResearchArtifact.zip