*Journal of Industrial and Management Optimization*

https://www.aimspress.com/journal/jimo

JIMO, 22(12): 6867–6907.
DOI:10.3934/jimo.2026229
Received: 01 September 2026
Revised: 15 November 2026
Accepted: 20 November 2026
Published: 25 November 2026

*Research article*

# Stochastic production-readiness design and order-acceptance scheduling in high-mix low-volume manufacturing

**Youngchul Kim[1,2], Chaehwa Lee[3], Hyeyoung Hwang[4], Jinho Cha[5] and Hongsuk Park[6,*]**

[1] Department of Industrial Engineering, Kumoh National Institute of Technology, Gumi, Republic of Korea
[2] K-Defense Industry Support Center, Konyang University, Daejeon, Republic of Korea
[3] Republic of Korea Army Training and Doctrine Command, Republic of Korea
[4] Republic of Korea Army, Seoul, Republic of Korea
[5] Department of Computer Science, Gwinnett Technical College, Lawrenceville, GA, USA
[6] Industry–Academic Cooperation Foundation, Kumoh National Institute of Technology, Gumi, Republic of Korea

* **Correspondence:** Email: avipak@kumoh.ac.kr.

**Abstract:** High-mix low-volume manufacturing requires production readiness under uncertain orders and limited capacity. We formulate a two-stage stochastic model that selects product–machine readiness before demand and then jointly accepts and schedules orders with alternative machines, sequence-dependent setups, and deadlines. We establish complexity and structural results and develop exact component-wise logic-based Benders decomposition and a capability-aware matheuristic. In the focal study, 20 sample average approximation replications select the same network; the stochastic design achieves out-of-sample net value 145.89 versus 65.29 for a deterministic-medoid benchmark. Component-wise decomposition reduces master iterations from 47 to 20. Across 48 additional instances, the schedule-aware heuristic improves mean net value by 28.99 over an aggregate proxy (95% paired interval [14.89,43.09]). However, a small-instance exact-recourse check finds a ranking reversal, so heuristic gains do not establish exact-policy superiority.

**Keywords:** high-mix low-volume manufacturing; production readiness; order acceptance and scheduling; stochastic programming; logic-based Benders decomposition
**Mathematics Subject Classification:** 90B35, 90C11, 90C15, 90C27

## 1. Introduction

High-mix low-volume (HMLV) manufacturing combines broad product variety with small, irregular, and value-heterogeneous orders. The setting appears in customized manufacturing, aerospace and industrial replacement parts, rail and energy-service components, legacy-equipment support, and defense sustainment. Flexible job-shop scheduling (FJSP) is a natural operational foundation because operations may use alternative machines and must still be sequenced feasibly [1]. Yet standard FJSP models normally take the available eligibility network as given; even models with alternative routes or process plans choose among alternatives that already exist [2]. HMLV systems face an additional decision: which technically feasible product–machine relations should be kept operationally ready before the future order portfolio is known.

A technically capable machine may still require a maintained computer numerical control program, fixture, tool set, inspection plan, qualification status, process documentation, or trained operator before it can process a product family. Maintaining every feasible relation can therefore be costly or administratively impossible, whereas an overly sparse readiness network can remove valuable routing options when urgent or high-value orders arrive. This decision differs from choosing a route after demand is observed: Readiness design chooses which alternatives remain available to the downstream schedule.

The second decision layer is order acceptance and scheduling (OAS). OAS makes acceptance conditional on whether accepted work can actually be assigned and scheduled [3]. Under stochastic arrivals and sequence-dependent setups, remaining capacity has opportunity cost [4]; in on-demand production, acceptance, assignment, and hard delivery commitments can be tightly coupled [5]. We therefore model order acceptance as recourse rather than as a fixed input. Before uncertainty is revealed, the planner selects a readiness network. After a scenario is realized, the planner accepts or rejects orders, assigns every accepted operation to a ready machine, and builds a sequence-dependent detailed schedule satisfying hard delivery constraints.

The contribution is positioned at the intersection of several established streams: Endogenous qualification, qualification with scheduling, production planning with order acceptance, eligibility-constrained OAS, and pre-demand process flexibility all have direct precedents [6–11]. Section 2 therefore focuses on the specific integration and decision resolution that distinguish the present model.

The paper makes four contributions. First, it formulates this coupling as a two-stage stochastic mixed-integer program in which first-stage readiness decisions determine the feasible set of detailed recourse. Second, it establishes strong NP-hardness, recourse monotonicity under off-line readiness, failure of that monotonicity under readiness-induced capacity loss, non-submodularity of readiness value, insufficiency of aggregate-capacity reasoning, value–volume reversal, and technical-component decomposition. Third, it derives readiness-cover, workload, energetic, setup-aware, scheduling-conflict, and component-wise logic-based Benders inequalities. Fourth, it develops an exact component-wise logic-based Benders decomposition (C–LBBD) and a capability-aware fix-and-optimize (CA–FO) matheuristic, and evaluates both on a synthetic benchmark with independently generated out-of-sample scenarios.

## 2. Related literature and research positioning

This study lies at the intersection of process flexibility, product–machine qualification, order acceptance and scheduling (OAS), and stochastic make-to-order production. The review focuses on the decision-resolution gap that remains when readiness must be committed before an uncertain HMLV order portfolio is known.

### *2.1. Readiness, flexibility, and qualification*

FJSP and process-flexibility research establish alternative-machine assignment and sparse product–resource networks as standard sources of operational flexibility [1, 2, 12, 13]. Related studies extend FJSP formulations toward transportation-aware processing and multiple time constraints [14, 15], while dynamic expedited-order scheduling under machine–tool compatibility has also been studied [16]. Parallel-machine flexible job-shop heuristics provide an additional scheduling precedent [17]. These contributions enrich downstream assignment and sequencing, but they do not make the available processing-capability network a pre-demand stochastic design variable. Qualification research moves the decision upstream. Fu et al. [6] and Chang and Dong [7] optimize qualification portfolios under uncertain demand and capacity effects, while Rowshannahad et al. [18] and Perraudat et al. [19, 20] study capacitated or robust qualification choices. Thus, endogenous qualification under uncertainty is already well established.

Qualification and detailed scheduling are also established. Li et al. [8, 21] jointly model production and qualification schedules; Tamssaouet and Dauzère-Pérès [22] make operation flexibility a decision in FJSP; and Nattaf et al. [23] study qualification loss during scheduling. Chang and Dong [24] further show that qualification can consume production capacity. Accordingly, the contribution does not rest on qualification itself, its integration with scheduling, or qualification-induced capacity loss. The base model instead treats readiness preparation as an off-line first-stage commitment; capacity loss is retained as a structural extension.

### *2.2. Order acceptance, stochastic production, and HMLV uncertainty*

OAS already couples accept/reject decisions with operational feasibility. Xu et al. [4] study stochastic arrivals with sequence-dependent setups, Li et al. [5] integrate acceptance, assignment, and hard due dates, and recent studies add eligibility and heterogeneous production resources, and have also applied logic-based Benders decomposition (LBBD) [10, 25, 26]. Thus, detailed OAS with fixed eligibility, sequence-dependent setup scheduling, and generic LBBD methodology are well established.

Three recent contributions further sharpen this boundary. Zheng et al. [27] analyze online single-machine acceptance, where orders arrive sequentially and acceptance must respond to partial information; our second stage instead observes the complete realized portfolio before acceptance and scheduling. Ervural and Özaydın [28] jointly optimize order acceptance, production, and inventory in a hybrid make-to-order/make-to-stock system, including a plant-based application, whereas our model has no inventory replenishment decision and endogenizes a pre-demand machine-readiness network. Xu et al. [29] provide an exact LBBD framework for order acceptance and non-permutation flow-shop scheduling. Their work is an important methodological precedent: Our novelty is not LBBD itself but readiness-network commitment, component-specific bounds, and the associated alternative-machine

HMLV recourse structure. These contrasts delimit the information timing, application scope, and solution contribution rather than suggesting that any individual ingredient is new.

Upstream production-planning models also make acceptance and flexibility decisions endogenous. Aouam and Brahimi [30] and Aouam et al. [9] integrate production planning and order acceptance, while Boulaksil et al. [31], Bish et al. [32], and Tanrisever et al. [11] establish pre-demand capacity/process-flexibility choices followed by adaptive production decisions. These papers provide the closest timing precedent, but their recourse is more aggregate than operation-level alternative-machine scheduling with sequence-dependent setups and hard delivery. HMLV and spare-parts evidence motivates sparse, lumpy, heterogeneous, and correlated order portfolios [33–35]; those studies discipline scenario generation here rather than add an inventory or forecasting decision to the model.

### *2.3. Integrated positioning*

The literature therefore establishes direct precedents for each individual ingredient. Within the reviewed literature, we did not identify a direct predecessor that simultaneously combines

> pre-demand binary product–machine readiness design
> + uncertain value-heterogeneous HMLV order portfolios
> + scenario-specific discretionary order acceptance
> + operation-level alternative-machine assignment and sequence-dependent setup scheduling.

Accordingly, the contribution is the integrated decision structure rather than any individual modeling ingredient. Table 1 summarizes this research boundary.

**Table 1.** Research positioning relative to qualification, HMLV scheduling, order acceptance, and stochastic flexibility.

| Research stream | Representative studies | Existing optimization lens | Gap relative to this study |
|---|---|---|---|
| Process flexibility and FJSP | [1, 2, 12–16] | Sparse product–resource flexibility, alternative routing, assignment, dynamic-order handling, and detailed scheduling. | Available flexibility is generally fixed before scheduling; no stochastic discretionary readiness-network commitment. |
| Product–machine qualification under uncertainty | [6, 7, 18, 19, 24] | Qualification portfolios under uncertain demand/capacity, qualification cost, and capacity loss. | Endogenous qualification is well established; detailed scenario-specific order rejection and sequence-dependent detailed scheduling are not jointly represented. |
| Qualification and detailed scheduling | [8, 22, 23] | Qualification selection/loss integrated with detailed production or FJSP scheduling. | Qualification integrated with scheduling is established; the missing dimension is pre-demand commitment before a stochastic discretionary order portfolio. |
| Order acceptance and detailed scheduling | [3–5, 10, 25–27, 29] | Accept/reject decisions integrated with assignment, sequence, setups, eligibility, and hard delivery. | Eligibility/flexibility is typically fixed rather than a first-stage readiness-network decision. |
| Stochastic make-to-order capacity/process flexibility | [11, 28, 31, 32] | Pre-demand flexibility/capacity choices followed by adaptive production or acceptance decisions. | Closest timing precedent, but recourse is aggregate rather than operation-level alternative-machine sequence-dependent setup scheduling. |
| Integrated scheduling and decomposition | [36–40] | Structural decomposition, LBBD/combinatorial Benders, strengthened formulations, and large-neighborhood methods. | Methodological foundation; the contribution here is the problem-specific readiness structure and associated inequalities rather than generic LBBD. |
| HMLV / intermittent-demand evidence | [33–35, 41] | Sparse, lumpy, heterogeneous, and correlated low-volume demand. | Motivates stochastic HMLV portfolios but does not optimize readiness networks with detailed OAS. |
| Present study | – | Pre-demand binary product–machine readiness + uncertain value-heterogeneous orders + discretionary acceptance + operation assignment + sequence-dependent setup scheduling. | The reviewed literature does not contain a direct predecessor with this complete combination. |

## 3. Problem definition and mathematical formulation

### 3.1. Notation

Table 2 collects the notation used in the formulation, structural results, and decomposition.

**Table 2.** Core notation for the HMLV production-readiness model.

| Panel | Symbol | Definition |
|---|---|---|
| Sets | $\mathcal{F}, \mathcal{M}, \Omega$ | Product families, machines, and scenarios |
| Sets | $O_\omega, \mathcal{R}_o$ | Orders realized in scenario $\omega$ and ordered operations of order $o$ |
| Sets | $\mathcal{U}_{m\omega}$ | Operations technically eligible for machine $m$ in scenario $\omega$ |
| Parameters | $\bar{e}_{fm}, e_{orm}$ | Technical family–machine and operation–machine feasibility |
| Parameters | $F_{fm}, b_{fm}, B$ | Fixed readiness cost, readiness-resource use, and readiness-resource budget |
| Parameters | $H^0_{m\omega}, \ell_{fm}$ | Base machine horizon and optional readiness-induced capacity loss |
| Parameters | $p_{orm\omega}, c_{orm\omega}$ | Processing time and variable processing cost |
| Parameters | $s_{ff'm}, c^S_{ff'm}$ | Sequence-dependent setup time and setup cost |
| Parameters | $r_{o\omega}, d_{o\omega}, v_{o\omega}$ | Release time, hard due date, and order value |
| Parameters | $\pi_\omega$ | Scenario probability |
| First stage | $z_{fm}$ | 1 if family–machine edge $(f, m)$ is maintained production-ready |
| Recourse | $a_{o\omega}$ | 1 if order $o$ is accepted |
| Recourse | $x_{orm\omega}$ | 1 if operation $(o, r)$ is assigned to machine $m$ |
| Recourse | $S_{or\omega}, C_{or\omega}$ | Operation start and completion times |
| Recourse | $y_{uvm\omega}$ | 1 if operation $v$ immediately follows operation $u$ on machine $m$ |
| Recourse | $\alpha_{um\omega}, \beta_{um\omega}$ | First- and last-operation indicators on machine $m$ |
| Recourse | $q_{fm\omega}, h_{m\omega}$ | Auxiliary family-use indicator for strengthening and core machine-use indicator |
| LBBD | $\theta_{c\omega}$ | Master estimate of recourse value for technical component $c$ in scenario $\omega$ |
| Value | $Q(z, \omega)$ | Optimal detailed recourse value under readiness network $z$ |

### 3.2. Problem setting and decision timing

**Technical feasibility versus operational readiness** Let $\mathcal{F}$ be the set of product families and $\mathcal{M}$ the set of machines. For each family–machine pair, let

$$\bar{e}_{fm} = \begin{cases} 1, & \text{if machine } m \text{ is technically capable of family } f, \\ 0, & \text{otherwise.} \end{cases}$$

Technical feasibility is a physical or engineering property and is treated as data.

The first-stage binary decision is

$$z_{fm} = \begin{cases} 1, & \text{if the family–machine pair } (f, m) \text{ is maintained production-ready,} \\ 0, & \text{otherwise.} \end{cases}$$

The effective readiness relation is therefore

$$e^{\mathrm{eff}}_{fm} = \bar{e}_{fm} z_{fm}.$$

Let $F_{fm}$ denote the monetary or value-equivalent cost of maintaining readiness on $(f, m)$ over the planning horizon, and let $b_{fm}$ denote the corresponding consumption of a limited readiness resource (e.g., qualification effort, fixture/tooling slots, certification workload, or readiness budget units). A readiness-resource limit $B$ captures the fact that not all technically possible links can necessarily be maintained. Qualification activities themselves are assumed completed before the operational recourse horizon; unlike Li et al. [8], their detailed timing is not a second scheduling layer in the present base model.

**Scenario-dependent order portfolios** Let $\omega \in \Omega$ denote an order-portfolio scenario with probability $\pi_\omega$. Scenario $\omega$ contains a set of orders $\mathcal{O}_\omega$. Order $o$ belongs to family $f(o)$, has release time $r_{o\omega}$, hard due date $d_{o\omega}$, and value $v_{o\omega}$. Order $o$ consists of a precedence-ordered set of operations $\mathcal{R}_o$. For operation $r \in \mathcal{R}_o$, $p_{orm\omega}$ is its processing time on machine $m$ and $e_{orm} \in \{0, 1\}$ indicates technical operation–machine feasibility. Sequence-dependent setup time between two operations of families $f$ and $f'$ on machine $m$ is $s_{ff'm}$.

At the second-stage decision epoch, the entire realized order portfolio $\mathcal{O}_\omega$, including the offered orders, releases, due dates, values, and operation requirements, is assumed observable before any acceptance, assignment, or sequencing decision is taken. This is a batch, two-stage recourse abstraction, not a sequential-arrival online OAS policy; any real-time order acceptance extension would require a different information structure and nonanticipativity constraints.

An order is accepted only if every required operation can be assigned and completed by its hard due date. Thus, the second-stage problem determines acceptance, machine assignment, and sequencing conditional on the first-stage readiness network.

**Modeling assumptions and decision timing** The theoretical results below use the following base assumptions unless explicitly stated otherwise.

**A1.** Technical feasibility is exogenous, with $e_{orm} \le \bar{e}_{f(o)m}$; readiness $z_{fm}$ is selected before scenario $\omega$ is observed.

**A2.** In the base model, readiness preparation is completed outside the operational scheduling horizon. Hence adding a ready edge cannot reduce machine time available to the second-stage schedule.

**A3.** The readiness state is fixed during each base-model recourse horizon. Qualification decay or disqualification during scheduling, as studied by Nattaf et al. [23], is outside the base model used to establish monotonicity.

**A4.** Rejecting all orders is feasible in every scenario.

**A5.** Feasible processing times are strictly positive; order values, readiness costs/resources, setup times, and processing/setup costs are nonnegative. Initial setup, when relevant, is absorbed into processing time/cost or normalized to zero.

**A6.** An accepted order must assign every required operation exactly once and satisfy release, precedence, machine-horizon, and hard delivery constraints.

**A7.** Scenario probabilities satisfy $\pi_\omega \geq 0$ and $\sum_{\omega\in\Omega} \pi_\omega = 1$ for the finite stochastic formulation. Training and out-of-sample scenario generation are described in Section 6.

The off-line-readiness assumption is relaxed in Section 3.3. This separation is intentional: Qualification-induced capacity loss is established in the qualification literature [24] and should not be hidden inside a generic fixed readiness cost.

### *3.3. Two-stage stochastic formulation*

Let $u = (o, r)$ denote an operation, let $f(u) = f(o)$, and use $e_{um}$, $p_{um\omega}$, and $c_{um\omega}$ as shorthand for the corresponding operation-indexed quantities. The formulation below permits pair-specific valid big-$M$ values. In the reported implementation, exact route/time-window and sequencing-arc preprocessing is applied first, after which a safe horizon-based big-$M$ constant is used; Section 5 gives the implementation details.

The first-stage problem is

$$\max_{z} \; -\sum_{f\in\mathcal{F}} \sum_{m\in\mathcal{M}} F_{fm} z_{fm} + \sum_{\omega\in\Omega} \pi_\omega Q(z,\omega) \tag{3.1}$$

subject to

$$\sum_{f\in\mathcal{F}} \sum_{m\in\mathcal{M}} b_{fm} z_{fm} \leq B, \tag{3.2}$$

$$z_{fm} \leq \bar{e}_{fm} \qquad \forall f, m, \tag{3.3}$$

and $z_{fm} \in \{0, 1\}$.

For fixed $z$ and scenario $\omega$, recourse value $Q(z, \omega)$ is

$$Q(z,\omega) = \max \left\{ \sum_{o\in\mathcal{O}_\omega} v_{o\omega} a_{o\omega} - \sum_{o,r,m} c_{orm\omega} x_{orm\omega} - \sum_{u\neq v} \sum_{m} c^{S}_{f(u)f(v)m} y_{uvm\omega} \right\}, \tag{3.4}$$

subject to the following core constraints.

Every operation of an accepted order is assigned exactly once:

$$\sum_{m\in\mathcal{M}} x_{orm\omega} = a_{o\omega} \qquad \forall o \in \mathcal{O}_\omega,\ r \in \mathcal{R}_o. \tag{3.5}$$

Assignment is possible only through a technically feasible and maintained-ready edge:

$$x_{orm\omega} \leq e_{orm}\, z_{f(o)m} \qquad \forall o, r, m. \tag{3.6}$$

Equation (3.6) is the defining coupling constraint of the paper.

**Numerical illustration of eligibility and readiness** Consider one family $f$ with two operations and two machines $m_1, m_2$. Both machines are technically capable of at least one operation of $f$, so $\bar{e}_{fm_1} = \bar{e}_{fm_2} = 1$. The first operation can use only $m_1$, whereas the second can use either machine: $e_{o1m_1} = 1$, $e_{o1m_2} = 0$, and $e_{o2m_1} = e_{o2m_2} = 1$. If readiness decisions are $z_{fm_1} = 0$ and $z_{fm_2} = 1$, the first operation has no usable machine, so the entire order must be rejected ($a_{o\omega} = 0$), even though machine $m_2$ is family-ready. Setting $z_{fm_1} = 1$ would restore technical assignment possibilities, but would not by itself guarantee a schedule meeting the due date. This example distinguishes technical family capability, operation eligibility, and maintained readiness.

Completion times satisfy

$$C_{or\omega} = S_{or\omega} + \sum_{m \in \mathcal{M}} p_{orm\omega} x_{orm\omega}, \qquad \forall o, r. \tag{3.7}$$

Release and within-order precedence constraints are

$$S_{or\omega} \geq r_{o\omega} a_{o\omega} \qquad \forall o, r, \tag{3.8}$$

$$S_{o,r+1,\omega} \geq C_{or\omega} \qquad \forall o,\ r, r+1 \in \mathcal{R}_o. \tag{3.9}$$

To represent sequence-dependent setups without overcounting, $y_{uvm\omega}$ denotes an *immediate-successor* arc rather than a generic pairwise precedence relation. Let $\mathcal{U}_{m\omega} = \{u : e_{um} = 1\}$ be the operations technically eligible for machine $m$ in scenario $\omega$. Binary variables $\alpha_{um\omega}$ and $\beta_{um\omega}$ identify the first and last assigned operations, and $h_{m\omega}$ indicates whether machine $m$ is used. The path constraints are

$$\sum_{u \in \mathcal{U}_{m\omega}} \alpha_{um\omega} = h_{m\omega}, \qquad \forall m, \tag{3.10}$$

$$\sum_{u \in \mathcal{U}_{m\omega}} \beta_{um\omega} = h_{m\omega}, \qquad \forall m, \tag{3.11}$$

$$\alpha_{um\omega} + \sum_{v \in \mathcal{U}_{m\omega} \setminus \{u\}} y_{vum\omega} = x_{um\omega}, \qquad \forall u, m, \tag{3.12}$$

$$\beta_{um\omega} + \sum_{v \in \mathcal{U}_{m\omega} \setminus \{u\}} y_{uvm\omega} = x_{um\omega}, \qquad \forall u, m, \tag{3.13}$$

$$h_{m\omega} \leq \sum_{u \in \mathcal{U}_{m\omega}} x_{um\omega} \leq |\mathcal{U}_{m\omega}| h_{m\omega}, \qquad \forall m. \tag{3.14}$$

For every feasible arc $u \neq v$ on machine $m$, temporal consistency requires

$$S_{v\omega} \geq C_{u\omega} + s_{f(u)f(v)m} - M_{uvm\omega}\,(1 - y_{uvm\omega})\,. \tag{3.15}$$

Because feasible processing times are strictly positive, these temporal constraints eliminate directed cycles among selected successor arcs. Pair-specific $M_{uvm\omega}$ values are obtained from valid release- and due-date bounds; impossible arcs are deleted before optimization.

The operational horizon is enforced both on individual completion times and on total machine occupation:

$$C_{u\omega} \leq H^0_{m\omega} + M^H_{um\omega}(1 - x_{um\omega}) \qquad \forall u, m. \tag{3.16}$$

$$\sum_{u} p_{um\omega} x_{um\omega} + \sum_{u \neq v} s_{f(u)f(v)m} y_{uvm\omega} \leq H^0_{m\omega} \qquad \forall m. \tag{3.17}$$

The second term in (3.17) charges setup time only on realized immediate-successor transitions, which is consistent with the setup-cost term in (3.4).

Let $r_o^{\text{last}}$ be the final operation of order $o$. Hard delivery commitments are enforced by

$$C_{o,r_o^{\text{last}},\omega} \le d_{o\omega} + M_o(1 - a_{o\omega}) \qquad \forall o. \tag{3.18}$$

This hard-due-date acceptance interpretation follows the operational logic used in on-demand OAS: If a feasible schedule cannot complete the order by its commitment, the order is rejected rather than scheduled tardily [5].

The core recourse domains are

$$a_{o\omega}, x_{um\omega}, y_{uvm\omega}, \alpha_{um\omega}, \beta_{um\omega}, h_{m\omega} \in \{0, 1\}, \qquad S_{u\omega}, C_{u\omega} \ge 0.$$

The auxiliary family-use variables $q_{fm\omega}$ are introduced explicitly with their linking constraints in Section 4.3; they are not needed to define the base recourse feasible set.

**Capacity-loss extension** To represent environments in which qualification or readiness maintenance consumes operational machine time, define the available machine horizon

$$H_{m\omega}(z) = H^0_{m\omega} - \sum_{f \in \mathcal{F}} \ell_{fm} z_{fm}, \qquad \ell_{fm} \ge 0. \tag{3.19}$$

In this extension, $H^0_{m\omega}$ in (3.16)–(3.17) is replaced by $H_{m\omega}(z)$. In particular,

$$C_{u\omega} \le H_{m\omega}(z) + M^H_{um\omega}(1 - x_{um\omega}), \tag{3.20}$$

$$\sum_u p_{um\omega} x_{um\omega} + \sum_{u \ne v} s_{f(u)f(v)m} y_{uvm\omega} \le H_{m\omega}(z). \tag{3.21}$$

The base model is recovered when $\ell_{fm} = 0$ for all $(f, m)$. The extension is included to examine how readiness-induced capacity consumption changes the structural properties of the base model; the underlying qualification-capacity tradeoff is established in prior work [24].

## 4. Structural properties and valid inequalities

### *4.1. Complexity and structural properties*

The results in this section establish the computational hardness, structural behavior, and valid relaxations that support the formulation and the solution methods.

#### Strong NP-hardness

**Theorem 4.1** (Strong NP-hardness)**.** *The proposed readiness-design and order-acceptance scheduling problem is strongly NP-hard, even in the restricted deterministic case with a single scenario, one operation per order, identical parallel machines, zero setup times and setup costs, zero variable processing costs, zero readiness costs, and all readiness links fixed active.*

*Proof.* We reduce from 3-Partition. Consider an instance with integers $a_1, \ldots, a_{3m}$ and $B$ satisfying

$$\sum_{i=1}^{3m} a_i = mB, \qquad \frac{B}{4} < a_i < \frac{B}{2} \quad \forall i.$$

Construct one scenario with $m$ identical machines and $3m$ orders. Order $i$ contains one operation, has processing time $p_i = a_i$ on every machine, release time zero, common hard due date $B$, and value $v_i = 1$. Every machine is technically feasible and already ready for every order; all setup and processing costs are zero. Hence the objective is simply the number of accepted orders.

If the 3-Partition instance has a feasible partition, assign each triple to one machine. Every machine receives total processing time exactly $B$, so all $3m$ orders complete by the common due date and the objective value is $3m$.

Conversely, suppose the constructed scheduling instance attains objective value $3m$. Then all orders are accepted and must fit on the $m$ machines by time $B$. Since total processing load equals $mB$, every machine must carry load exactly $B$. Because every $a_i > B/4$, four orders cannot fit on one machine. Because every $a_i < B/2$, one or two orders cannot have total load exactly $B$. Hence every machine contains exactly three orders whose processing times sum to $B$, yielding a valid 3-Partition. The construction is polynomial, and 3-Partition is strongly NP-complete. Therefore, the restricted scheduling problem, and hence the proposed integrated problem, is strongly NP-hard. □

**Monotonicity under off-line readiness** For a readiness-edge set $S \subseteq \{(f, m) : \bar{e}_{fm} = 1\}$, let $Q(S, \omega)$ denote the optimal second-stage value in scenario $\omega$ under the base assumptions.

**Proposition 4.2** (Scenario-wise recourse monotonicity)**.** *Under the base assumptions A1–A7 with $\ell_{fm} = 0$, if $S \subseteq T$, then*

$$Q(S, \omega) \leq Q(T, \omega) \qquad \forall \omega.$$

*Consequently,*

$$\mathbb{E}_\omega[Q(S, \omega)] \leq \mathbb{E}_\omega[Q(T, \omega)].$$

*Proof.* Take an arbitrary feasible second-stage solution under readiness set $S$. Because $S \subseteq T$ and readiness does not consume operational capacity in the base model, every product–machine edge available under $S$ is also available under $T$. All acceptance, assignment, sequencing, precedence, release, and due-date constraints satisfied under $S$ therefore remain satisfied under $T$. Thus, the feasible region under $S$ is a subset of the feasible region under $T$. Maximization over the larger feasible region cannot yield a smaller recourse value. Taking expectations preserves the inequality. □

**Why capacity-aware qualification changes the theorem**

**Proposition 4.3** (Monotonicity may fail with readiness-induced capacity loss)**.** *If Equation* (3.19) *is active with some $\ell_{fm} > 0$, there exist instances with $S \subset T$ for which*

$$Q(T, \omega) < Q(S, \omega).$$

*Proof.* Consider one machine with initial available horizon $H^0 = 10$ and one order of value 1 requiring one operation of processing time 9, release time 0, and due date 10. Set all operational costs to zero. Let

the readiness edge required by this order belong to $S$ and have zero capacity loss. Then the order is feasible and $Q(S, \omega) = 1$.

Now add an otherwise unused readiness edge $e$ on the same machine, belonging to a family with no realized order in this scenario, and set $\ell_e = 2$. Choose the readiness-resource budget large enough that both $S$ and $T = S \cup \{e\}$ are first-stage feasible. Under $T$, the available machine horizon becomes 8. The 9-unit operation can no longer complete within the operational horizon, so rejecting the order is optimal and $Q(T, \omega) = 0$. Hence adding a readiness edge can reduce recourse value once readiness consumes operational capacity. □

This proposition formalizes the modeling implication of qualification-induced capacity loss emphasized by Chang and Dong [24]: "more flexibility" is monotone only when readiness is genuinely off-line with respect to the operational horizon.

**Aggregate capacity does not guarantee schedule feasibility**

**Proposition 4.4** (Aggregate-capacity insufficiency)**.** *Satisfaction of aggregate processing-capacity constraints is not sufficient for feasibility of the detailed HMLV schedule when sequence-dependent setups are present.*

*Proof.* Consider one machine available on $[0, 10]$ and two accepted single-operation orders from different families, both released at time zero with hard due date 10. Each operation has processing time 4, initial setup is zero, and the setup time between the two families is 3 in either direction. The aggregate processing load is

$$4 + 4 = 8 \leq 10,$$

so an aggregate capacity model declares the workload feasible. In any detailed sequence, however, total machine occupation is

$$4 + 3 + 4 = 11 > 10.$$

Hence no feasible detailed schedule exists. Therefore, aggregate capacity sufficiency does not imply schedule feasibility. □

The counterexample motivates an aggregate, schedule-ignorant benchmark in the computational study and clarifies why capacity feasibility alone cannot certify the detailed recourse schedule [18, 19].

**Readiness complementarity and non-submodularity** Define

$$V(S) = \mathbb{E}_\omega[Q(S, \omega)]$$

for the base model.

**Proposition 4.5** (Readiness value is not generally submodular)**.** *The set function $V(S)$ need not be submodular.*

*Proof.* Consider a deterministic one-scenario instance with one order of value 1 and two precedence-related operations $u_1 \prec u_2$. Operation $u_1$ can be processed only on machine $m_1$, and operation $u_2$ only on machine $m_2$. Both operations belong to family $f$, and the corresponding readiness edges are

$$e_1 = (f, m_1), \qquad e_2 = (f, m_2).$$

Let the due date be sufficiently large and all processing/setup costs be zero. The order can be accepted if and only if both readiness edges are active. Therefore,

$$V(\varnothing) = V(\{e_1\}) = V(\{e_2\}) = 0, \qquad V(\{e_1, e_2\}) = 1.$$

Take $A = \varnothing$, $B = \{e_1\}$, and $e = e_2$. Then

$$V(A \cup \{e\}) - V(A) = 0,$$

whereas

$$V(B \cup \{e\}) - V(B) = 1.$$

This violates the diminishing-returns inequality required for submodularity. Hence $V$ is not generally submodular. □

The counterexample also shows why independent edge ranking can be misleading: An edge with zero standalone value can have positive marginal value once a complementary readiness edge is present. This complements, rather than contradicts, the supermodularity phenomena established for particular long-chain flexibility structures by Simchi-Levi and Wei [13]; our detailed OAS recourse induces a different set function.

**Demand-volume readiness and value-aware readiness can reverse**

**Proposition 4.6** (Order value can reverse a volume-based readiness choice)**.** *There exist instances in which a readiness portfolio chosen to maximize covered demand volume is strictly suboptimal for the value-aware OAS objective. Both its absolute value loss and its approximation ratio relative to the value-aware optimum can be made arbitrarily large.*

*Proof.* Let the readiness resource permit exactly one of two edges, $e_A$ or $e_B$, and set both readiness costs to zero. Family $A$ has two feasible one-unit orders, each of value 1; family $B$ has one feasible one-unit order of value $M > 2$. All orders are otherwise schedulable without setup or congestion. A volume-based policy selects $e_A$ because it covers two demand units rather than one, obtaining value 2. The value-aware policy selects $e_B$ and obtains value $M$. The absolute loss is $M - 2$, while the ratio between the value-aware optimum and the volume-based value is $M/2$; both are unbounded as $M$ grows. Thus, demand volume alone cannot provide a bounded approximation guarantee for the value-aware readiness problem under unrestricted value heterogeneity. □

**Conservative global readiness screening** Let

$$\overline{V} = \mathbb{E}_\omega \left[ \sum_{o \in O_\omega} v_{o\omega} \right].$$

**Proposition 4.7** (Global cost screening)**.** *Under Assumptions A4–A5, if a readiness edge e has fixed cost $F_e \geq \overline{V}$, then there exists an optimal first-stage solution with $z_e = 0$.*

*Proof.* For every readiness set and scenario, reject-all feasibility and nonnegative operational costs imply

$$0 \le Q(S,\omega) \le \sum_{o\in O_\omega} v_{o\omega}.$$

Take any feasible first-stage readiness set $S$ containing $e$ and let $S^- = S \setminus \{e\}$. Removing $e$ preserves first-stage feasibility because it weakly relaxes the readiness-resource constraint. Moreover,

$$Q(S,\omega) - Q(S^-,\omega) \le \sum_{o\in O_\omega} v_{o\omega},$$

since $Q(S^-,\omega) \ge 0$. Therefore, if $\Phi(S)$ denotes the complete first-stage objective,

$$\Phi(S^-) - \Phi(S) = F_e + \mathbb{E}_\omega[Q(S^-,\omega) - Q(S,\omega)] \ge F_e - \overline{V} \ge 0.$$

Thus, every feasible solution containing $e$ has a no-worse counterpart without $e$, and at least one optimal solution satisfies $z_e = 0$. □

The bound is intentionally conservative and is used only as a safe preprocessing rule, not as a tight approximation of marginal readiness value. Stronger preprocessing is applied only when it follows from exact route or time-window feasibility logic.

**Technical-component decomposition and component-wise screening** Let

$$\mathcal{G} = (\mathcal{F} \cup \mathcal{M}, \bar{\mathcal{E}}), \qquad \bar{\mathcal{E}} = \{(f,m) : \bar{e}_{fm} = 1\},$$

be the bipartite graph of technical family–machine feasibility. Let $C$ denote its connected components. For component $c$, write $\mathcal{F}_c$, $\mathcal{M}_c$, and $\bar{\mathcal{E}}_c$ for its family, machine, and edge sets.

**Theorem 4.8** (Scenario recourse decomposes over technical components)**.** *Assume that (i) every order belongs to one family, (ii) operations of an order can be processed only on machines connected to that family in $\mathcal{G}$, (iii) processing and setup costs as well as second-stage machine capacities and setup constraints are machine local, and (iv) there is no second-stage resource or linking constraint coupling different connected components. Then, for every fixed readiness vector $z$ and scenario $\omega$,*

$$Q(z,\omega) = \sum_{c\in C} Q_c(z_c,\omega),$$

*where $z_c = (z_e : e \in \bar{\mathcal{E}}_c)$ and $Q_c$ is the optimal recourse value of component $c$.*

*Proof.* Because $\mathcal{G}$ is partitioned into connected components, a family in $\mathcal{F}_c$ is technically adjacent only to machines in $\mathcal{M}_c$, and no machine in $\mathcal{M}_c$ is adjacent to a family outside $\mathcal{F}_c$. Hence every assignment variable associated with an order from $\mathcal{F}_c$ uses a machine in $\mathcal{M}_c$. By assumptions (iii)–(iv), all sequencing, setup, machine-capacity, release, precedence, and due-date constraints involving such operations are contained within component $c$. The second-stage feasible region is therefore the Cartesian product of the component feasible regions. The second-stage objective is additive over accepted orders, processing assignments, and machine-local setups, so maximizing over that Cartesian product yields the sum of the component optima. □

Define the gross scenario value available in component $c$ as

$$U_{c\omega} = \sum_{o \in O_\omega:\, f(o) \in \mathcal{F}_c} v_{o\omega}, \qquad \bar{U}_c = \mathbb{E}_\omega[U_{c\omega}].$$

**Corollary 4.9** (Component-wise marginal-value bound)**.** *For any readiness edge $e \in \bar{\mathcal{E}}_c$ and any readiness set $S$ not containing $e$,*

$$0 \le \mathbb{E}_\omega[Q(S \cup \{e\}, \omega) - Q(S, \omega)] \le \bar{U}_c$$

*under the off-line-readiness assumptions.*

*Proof.* The lower bound follows from recourse monotonicity. By the decomposition theorem, changing edge $e$ cannot alter recourse decisions outside component $c$. Within component $c$, recourse value cannot increase by more than the total gross value of all orders in that component because operational costs are nonnegative. Taking expectations gives the result. □

**Corollary 4.10** (Component cost screening)**.** *If $e \in \bar{\mathcal{E}}_c$ satisfies*

$$F_e \ge \bar{U}_c,$$

*then there exists an optimal first-stage solution with $z_e = 0$.*

*Proof.* Removing $e$ saves $F_e$, weakly relaxes the readiness-resource constraint, and by the previous corollary can reduce expected recourse by at most $\bar{U}_c$. Thus, deleting $e$ cannot reduce total expected objective value when $F_e \ge \bar{U}_c$. □

This component bound weakly strengthens the global screening rule and is strictly tighter whenever $\bar{U}_c < \overline{V}$. The same decomposition also creates a natural parallelization unit for scenario subproblems.

### *4.2. Polyhedral strengthening and scheduling feasibility theory*

The strengthening framework draws on three classical strands of operations research methodology.

First, Gu et al. [42] study the practical use of lifted cover and lifted generalized upper-bound (GUB) cover inequalities in branch-and-cut for 0–1 integer programs. Their main message is computational as well as polyhedral: A minimal cover cut is only the starting point, and lifting variables outside the cover can substantially strengthen the relaxation when coefficient generation and separation are implemented carefully. Their companion complexity analysis [43] also provides an important warning: Separation over all lifted minimal covers is itself computationally hard, so a practical algorithm should use targeted cover generation and lifting rather than promise exhaustive separation. This directly motivates the readiness-budget lifting procedure developed below.

Second, Queyranne [44] characterizes the completion-time polyhedron of the single-machine nonpreemptive scheduling problem. For every nonempty job subset $S$, the completion-time inequality

$$\sum_{u \in S} p_u C_u \ge \frac{1}{2}\left[\left(\sum_{u \in S} p_u\right)^2 + \sum_{u \in S} p_u^2\right]$$

is facet defining for the simple scheduling polyhedron, and the resulting system admits polynomial-time separation. The precedence-constrained extension of Queyranne and Wang [45] shows that additional

scheduling structure can also be reflected polyhedrally. We use the simple completion-time inequality conservatively: If the latest completion bounds of an assignment subset already violate Queyranne's necessary inequality, that assignment subset is a valid scheduling conflict.

Third, Baptiste et al. [46] develop satisfiability tests and time-bound adjustments for cumulative scheduling, including energetic reasoning. Their required-energy construction measures the amount of processing that an activity must execute inside a time interval even when its entire time window is not contained in that interval. This construction is stronger than a test that accounts only for activities whose complete time windows lie inside the interval. In the unit-capacity machine subproblems, energetic reasoning therefore provides a stronger necessary condition before detailed sequencing is solved.

Lifted covers, Queyranne inequalities, and energetic reasoning are established methodology. Their role here is their specialization to the coupled readiness–acceptance–assignment structure together with implication cuts that connect first-stage readiness to scenario-specific HMLV workload.

### *4.3. Valid inequalities and problem-specific Benders cuts*

This section develops inequalities that are valid for the base model independently of any particular solver. Some are static master relaxations; others are generated dynamically from solved scheduling subproblems. Their role is to make the theoretical results computationally operative.

**Readiness knapsack-cover inequalities** Let $C \subseteq \bar{\mathcal{E}}$ be a readiness cover satisfying

$$\sum_{e \in C} b_e > B.$$

**Proposition 4.11** (Readiness cover inequality)**.** *Every feasible first-stage readiness vector satisfies*

$$\sum_{e \in C} z_e \leq |C| - 1. \tag{4.1}$$

*If $C$ is inclusion-minimal with respect to $\sum_{e \in C} b_e > B$, the inequality is a minimal knapsack-cover inequality.*

*Proof.* If $z_e = 1$ for every $e \in C$, then the readiness-resource consumption is at least $\sum_{e \in C} b_e > B$, violating the first-stage resource constraint. Hence at least one edge in $C$ must be inactive. □

The basic cover cut can be strengthened by sequentially lifting readiness variables outside $C$. Let $L$ be the set of variables already lifted and suppose

$$\sum_{e \in C} z_e + \sum_{j \in L} \lambda_j z_j \leq |C| - 1 \tag{4.2}$$

is valid on the restricted readiness knapsack in which all not-yet-lifted variables are fixed to zero. For an unlifted edge $k \notin C \cup L$ for which the restricted knapsack with $z_k = 1$ is nonempty, define

$$\lambda_k = |C| - 1 - \max \left\{ \sum_{e \in C} z_e + \sum_{j \in L} \lambda_j z_j : \begin{array}{l} \sum_e b_e z_e \leq B, \\ z_k = 1, \\ z_h = 0 \quad \forall h \in U \setminus \{k\}, \\ z \in \{0,1\}^{|\bar{\mathcal{E}}|} \end{array} \right\}, \tag{4.3}$$

where $U$ is the set of not-yet-lifted variables. If the restricted set with $z_k = 1$ is empty, then $z_k = 0$ is itself valid and the variable is fixed rather than lifted.

**Theorem 4.12** (Sequentially lifted readiness-cover inequality)**.** *If* (4.2) *is valid for the current restricted readiness knapsack, then*

$$\sum_{e\in C} z_e + \sum_{j\in L} \lambda_j z_j + \lambda_k z_k \le |C| - 1 \tag{4.4}$$

*is valid after variable $z_k$ is released. Repeating the procedure yields a valid lifted cover inequality for the full readiness knapsack.*

*Proof.* If $z_k = 0$, inequality (4.4) reduces to the valid inequality (4.2). If $z_k = 1$, the definition of $\lambda_k$ in (4.3) gives

$$\sum_{e\in C} z_e + \sum_{j\in L} \lambda_j z_j \le |C| - 1 - \lambda_k,$$

which is exactly (4.4). Thus, the inequality is valid for both values of $z_k$. Induction over the lifting sequence proves validity after all selected variables have been released. □

This is a direct specialization of sequential lifting for 0–1 knapsack covers [42]. Because exhaustive separation of lifted minimal covers is computationally difficult [43], the implementation separates only covers generated from fractional readiness solutions and then lifts a restricted candidate set.

**Family workload inequalities** For every operation $(o, r)$ having at least one technically feasible machine, define

$$\underline{p}_{or\omega} = \min\{p_{orm\omega} : e_{orm} = 1\}.$$

Any order containing an operation with no technically feasible machine is forced to $a_{o\omega} = 0$ by preprocessing and is excluded from the workload expressions below.

**Proposition 4.13** (Family readiness-capacity inequality)**.** *For every family $f$ and scenario $\omega$, every feasible recourse solution satisfies*

$$\sum_{o\in\mathcal{O}_\omega:\, f(o)=f} a_{o\omega} \sum_{r\in\mathcal{R}_o} \underline{p}_{or\omega} \le \sum_{m\in\mathcal{M}} H^0_{m\omega} z_{fm}. \tag{4.5}$$

*Proof.* Every accepted operation of family $f$ requires at least $\underline{p}_{or\omega}$ units of processing on a machine for which the family is ready. The total processing time available to family $f$ cannot exceed the sum of the full machine horizons over machines with $z_{fm} = 1$. Ignoring competition from other families and ignoring setup times can only enlarge the right-hand-side capacity available to family $f$, so the inequality is a necessary condition for detailed feasibility. □

A stronger implication can be obtained for a specified order subset. Let $\mathcal{O}' \subseteq \{o \in \mathcal{O}_\omega : f(o) = f\}$ and define its minimum processing workload

$$L_f(\mathcal{O}', \omega) = \sum_{o\in\mathcal{O}'} \sum_{r\in\mathcal{R}_o} \underline{p}_{or\omega}.$$

Sort the horizons of technically feasible machines for family $f$ as

$$H^{(1)}_{f\omega} \ge H^{(2)}_{f\omega} \ge \cdots \ge H^{(M_f)}_{f\omega},$$

and define

$$\kappa_f(O', \omega) = \min\left\{k : \sum_{j=1}^{k} H_{f\omega}^{(j)} \geq L_f(O', \omega)\right\},$$

whenever such a $k$ exists.

**Proposition 4.14** (Conditional readiness-cardinality inequality)**.** *For every family-$f$ order subset $O'$ for which $\kappa_f(O', \omega)$ is defined,*

$$\sum_{m:\bar{e}_{fm}=1} z_{fm} \geq \kappa_f(O', \omega)\left(\sum_{o\in O'} a_{o\omega} - |O'| + 1\right) \tag{4.6}$$

*is valid.*

*Proof.* If at least one order in $O'$ is rejected, the parenthetical term is nonpositive and (4.6) is trivial. If every order in $O'$ is accepted, the right-hand side equals $\kappa_f(O', \omega)$. Suppose fewer than $\kappa_f(O', \omega)$ family-$f$ readiness links were active. Even if those links corresponded to the largest available machine horizons, their total processing capacity would be strictly smaller than $L_f(O', \omega)$ by the definition of $\kappa_f$. This contradicts the minimum processing workload required by the accepted orders. Hence at least $\kappa_f(O', \omega)$ readiness links must be active. □

Inequality (4.6) is deliberately conservative because it ignores operation eligibility differences, setup loss, and competition from other families. Those omissions can only overestimate usable capacity, so they preserve validity while keeping separation inexpensive.

**Energetic-reasoning inequalities** Let $\underline{r}_{u\omega}$ be a valid lower bound on the start time of operation $u$ and let $\overline{d}_{u\omega}$ be a valid upper bound on its completion time whenever $u$ is accepted and assigned. For machine $m$, operation $u \in \mathcal{U}_{m\omega}$, and interval $I = [t_1, t_2]$, define the minimum amount of processing of $u$ that must occur inside $I$ if $u$ is assigned to $m$:

$$w_{um\omega}(I) = \max\left\{0,\ p_{um\omega} - \max(0, t_1 - \underline{r}_{u\omega}) - \max(0, \overline{d}_{u\omega} - t_2)\right\}. \tag{4.7}$$

**Proposition 4.15** (Machine energetic-reasoning inequality)**.** *For every machine $m$, scenario $\omega$, and nonempty interval $I = [t_1, t_2]$,*

$$\sum_{u\in\mathcal{U}_{m\omega}} w_{um\omega}(I) x_{um\omega} \leq t_2 - t_1. \tag{4.8}$$

*Proof.* Consider an assigned operation $u$ with $x_{um\omega} = 1$. At most $\max(0, t_1 - \underline{r}_{u\omega})$ units of its processing can occur before $t_1$, because the operation cannot start before $\underline{r}_{u\omega}$. Similarly, at most $\max(0, \overline{d}_{u\omega} - t_2)$ units can occur after $t_2$, because it must complete no later than $\overline{d}_{u\omega}$. Therefore, at least $w_{um\omega}(I)$ processing units must occupy machine $m$ inside $I$. Summing these mandatory amounts over all assigned operations and using unit machine capacity gives (4.8). □

The construction is the unit-capacity specialization of energetic reasoning for cumulative scheduling [46].

**Corollary 4.16** (Dominance over the fully-contained-window cut)**.** *Let*

$$\mathcal{J}_{m\omega}(I) = \{u \in \mathcal{U}_{m\omega} : \underline{r}_{u\omega} \geq t_1,\ \overline{d}_{u\omega} \leq t_2\}.$$

*Then* (4.8) *dominates the earlier inequality*

$$\sum_{u \in \mathcal{J}_{m\omega}(I)} p_{um\omega} x_{um\omega} \leq t_2 - t_1.$$

*Proof.* For every $u \in \mathcal{J}_{m\omega}(I)$, no processing can be placed outside $I$, so $w_{um\omega}(I) = p_{um\omega}$. For every other operation, the energetic coefficient is nonnegative, whereas the fully-contained cut assigns coefficient zero. Thus, the left-hand side of (4.8) is never smaller at any nonnegative assignment vector and coincides on the fully-contained operations. □

**Setup-aware machine-load inequality** Introduce $q_{fm\omega} \in \{0, 1\}$ to indicate that at least one operation of family $f$ is assigned to machine $m$. Together with the machine-use indicator $h_{m\omega}$ from Section 3, impose

$$x_{um\omega} \leq q_{f(u)m\omega}, \qquad q_{fm\omega} \leq \sum_{u:\, f(u)=f} x_{um\omega}, \qquad q_{fm\omega} \leq h_{m\omega}, \qquad h_{m\omega} \leq \sum_{f} q_{fm\omega}.$$

Let

$$\delta_m = \min\{s_{ff'm} : f \neq f',\ \bar{e}_{fm} = \bar{e}_{f'm} = 1\},$$

where the minimum is over ordered inter-family transitions technically relevant to machine $m$; set $\delta_m = 0$ if fewer than two families can use the machine. Thus, $\delta_m$ is zero whenever the relevant transition set has a zero-setup arc.

**Proposition 4.17** (Family-transition load inequality)**.** *Every feasible schedule satisfies*

$$\sum_{u \in \mathcal{U}_{m\omega}} p_{um\omega} x_{um\omega} + \delta_m \left( \sum_{f} q_{fm\omega} - h_{m\omega} \right) \leq H^0_{m\omega}. \tag{4.9}$$

*Proof.* If machine $m$ is unused, $h_{m\omega} = 0$ and the inequality is trivial. Otherwise let $k = \sum_f q_{fm\omega}$ be the number of distinct families processed on $m$. Any sequence visiting $k$ distinct families contains at least $k - 1$ inter-family transitions. Each such transition consumes at least $\delta_m$ setup time. Thus, total machine occupation is at least the processing load plus $\delta_m(k - 1)$, which cannot exceed the available horizon. □

**Queyranne-derived due-date conflict cuts** For a fixed machine $m$, scenario $\omega$, and nonempty subset $S \subseteq \mathcal{U}_{m\omega}$, define

$$g_m(S) = \frac{1}{2} \left[ \left( \sum_{u \in S} p_{um\omega} \right)^2 + \sum_{u \in S} p^2_{um\omega} \right]. \tag{4.10}$$

**Proposition 4.18** (Queyranne due-date conflict)**.** *If*

$$\sum_{u \in S} p_{um\omega} \overline{d}_{u\omega} < g_m(S), \tag{4.11}$$

*then the assignments* $\{x_{um\omega} = 1 : u \in S\}$ *cannot all occur in a feasible schedule. Hence*

$$\sum_{u \in S} x_{um\omega} \leq |S| - 1 \tag{4.12}$$

*is valid.*

*Proof.* Assume, to the contrary, that all operations in $S$ are assigned to machine $m$ in a feasible schedule. Ignoring release dates, precedence restrictions, and sequence-dependent setup times can only relax that schedule. For any nonpreemptive single-machine schedule, Queyranne's completion-time inequality gives

$$\sum_{u \in S} p_{um\omega} C_{u\omega} \geq g_m(S)$$

[44]. Feasibility also requires $C_{u\omega} \leq \overline{d}_{u\omega}$ for every $u \in S$. Therefore,

$$g_m(S) \leq \sum_{u \in S} p_{um\omega} C_{u\omega} \leq \sum_{u \in S} p_{um\omega} \overline{d}_{u\omega},$$

contradicting (4.11). Thus, at least one assignment in $S$ must be removed, proving (4.12). □

This cut is analytical rather than solver-generated: It can identify a scheduling conflict before the full disjunctive subproblem is solved. Positive setup times and additional precedence constraints do not invalidate the test because they only make the detailed schedule more restrictive.

**Minimal scheduling-conflict cuts** Fix a scenario $\omega$. A set $C \subseteq \{(u, m) : u \in \mathcal{U}_{m\omega}\}$ is called a scheduling conflict if fixing $x_{um\omega} = 1$ for every $(u, m) \in C$ makes the complete recourse problem infeasible, while all other recourse variables remain free.

**Proposition 4.19** (Combinatorial scheduling-conflict cut)**.** *For every scheduling conflict* $C$*,*

$$\sum_{(u,m) \in C} x_{um\omega} \leq |C| - 1. \tag{4.13}$$

*is valid. Inclusion-minimal conflicts avoid immediate dominance by a cut generated from a strict-subset conflict.*

*Proof.* If all assignments in $C$ were simultaneously equal to one, the detailed scheduling subproblem would be infeasible by definition of a conflict. Therefore, at least one assignment in $C$ must be removed. Minimality is not required for validity, but deleting any literal from an inclusion-minimal conflict destroys the same infeasibility certificate. □

This is the scheduling analogue of combinatorial Benders inequalities generated from minimal infeasible subsystems [40]. A deletion filter or a suitable conflict-refinement procedure can be used to shrink an infeasible assignment set before adding (4.13).

**Component-wise monotone outer optimality cuts** For component $c$ and scenario $\omega$, let $\bar{z}_c$ be a readiness vector already evaluated by the exact component scheduling recourse, with

$$\bar{Q}_{c\omega} = Q_c(\bar{z}_c, \omega), \qquad U_{c\omega} = \sum_{o: f(o) \in \mathcal{F}_c} v_{o\omega}.$$

Let $\theta_{c\omega}$ be the master variable representing component recourse value.

**Theorem 4.20** (Monotone component optimality cut)**.** *Under the technical-component decomposition conditions, the off-line-readiness assumptions, and A4–A5, the inequality*

$$\theta_{c\omega} \le \bar{Q}_{c\omega} + \left(U_{c\omega} - \bar{Q}_{c\omega}\right) \sum_{e \in \bar{\mathcal{E}}_c:\, \bar{z}_e = 0} z_e. \tag{4.14}$$

*is a valid upper-bounding logic-based Benders cut for the maximization master problem. It is sharp at* $z_c = \bar{z}_c$.

*Proof.* First suppose

$$\sum_{e: \bar{z}_e = 0} z_e = 0.$$

Then $z_c \le \bar{z}_c$ componentwise. Scenario-wise recourse monotonicity gives

$$Q_c(z_c, \omega) \le Q_c(\bar{z}_c, \omega) = \bar{Q}_{c\omega},$$

which is exactly the right-hand side of (4.14).

Now suppose at least one previously inactive edge is activated. The right-hand side is at least

$$\bar{Q}_{c\omega} + U_{c\omega} - \bar{Q}_{c\omega} = U_{c\omega}.$$

Because all operating costs are nonnegative, component recourse cannot exceed the gross value of all orders in that component, so

$$Q_c(z_c, \omega) \le U_{c\omega}.$$

Thus, the cut is valid in both cases. At $z_c = \bar{z}_c$, the summation term is zero and the cut gives $\theta_{c\omega} \le \bar{Q}_{c\omega}$, so it is sharp. □

The cut is stronger than an exact-vector nogood bound because it simultaneously bounds every readiness vector that is a subset of $\bar{z}_c$. It is problem specific: Its validity uses the monotonicity theorem and therefore does not extend unchanged to the capacity-loss variant.

**Finite convergence of the outer LBBD** Consider an outer master with variables $z$ and $\theta_{c\omega}$, the readiness-resource constraints, the initial valid bounds

$$0 \le \theta_{c\omega} \le U_{c\omega}, \qquad c \in C,\ \omega \in \Omega,$$

static inequalities such as (4.1), and all generated cuts (4.14). At each incumbent readiness vector, all component/scenario recourse problems are solved exactly.

**Theorem 4.21** (Finite convergence of exact outer LBBD)**.** *Under the base model assumptions and the technical-component decomposition conditions, if each master problem and every component/scenario recourse subproblem is solved to global optimality, the outer LBBD algorithm using valid cuts* (4.14) *terminates after finitely many evaluated readiness vectors with a globally optimal solution.*

*Proof.* The readiness domain is finite because $z$ is binary over the finite edge set $\bar{\mathcal{E}}$. Every generated cut is a valid upper bound on the corresponding component recourse and is exact at the component readiness vector that generated it. Because the master is a maximization problem, its optimal objective is therefore a valid global upper bound, while exact evaluation of any master readiness vector provides a feasible lower bound.

After a readiness vector $\bar{z}$ is evaluated, the newly added component cuts make the master recourse estimate exact at $\bar{z}$; no valid cut can reduce that estimate below the true recourse value. If a globally solved master selects an already evaluated readiness vector again, its master objective equals the feasible objective of that vector. The global upper and lower bounds then coincide, which certifies optimality. Otherwise a different binary readiness vector is selected. Since there are finitely many such vectors, the procedure must either certify optimality earlier or evaluate all feasible readiness vectors, after which the best evaluated vector is globally optimal. □

This is a problem-specific specialization of the finite-convergence logic underlying LBBD [38, 47]. The practical issue is therefore the strength of the master bounds and the cost of repeated exact recourse evaluation, which are examined computationally in Section 6.

**Scope of polyhedral claims** The inequalities above are stated at the level justified by their proofs. Validity is established for the readiness–scheduling model, and dominance is claimed only where one inequality is explicitly shown to imply another. No inequality newly derived for the coupled model is claimed to be facet defining. Facet results for the classical completion-time or knapsack polytopes [42, 44] do not automatically transfer to the coupled readiness–scheduling formulation.

## 5. Exact and scalable solution algorithms

### *5.1. Decomposition foundations and exact recourse architecture*

The exact method follows the logic-based Benders decomposition (LBBD) framework of Hooker and Ottosson [47] and the planning–scheduling specialization of Hooker [38]. The stochastic extension of Elçi and Hooker [39] and the combinatorial-conflict perspective of Codato and Fischetti [40] provide the relevant methodological precedents. We do not rely on generic LBBD, branch-and-check, or conflict cuts as methodological novelties. The contribution is the readiness-specific technical-component decomposition and the way its exact recourse bounds are coupled to the first-stage readiness network.

Let $C$ be the connected components of the technical family–machine graph. At an outer-LBBD iteration, the master contains the first-stage readiness variables and one recourse-estimate variable $\theta_{c\omega}$ for each component–scenario pair. With $\mathcal{K}$ denoting the set of generated component cuts, the master

has the form

$$\begin{aligned} \max_{z,\theta} \quad & -\sum_{f,m} F_{fm} z_{fm} + \sum_{\omega\in\Omega} \pi_\omega \sum_{c\in C} \theta_{c\omega}, \\ \text{s.t.} \quad & \sum_{f,m} b_{fm} z_{fm} \le B, \qquad z_{fm} \le \bar{e}_{fm}, \\ & 0 \le \theta_{c\omega} \le U_{c\omega}, \qquad c \in C,\ \omega \in \Omega, \\ & z_{fm} \in \{0,1\}, \\ & \text{master-valid readiness inequalities and all cuts in } \mathcal{K}. \end{aligned} \tag{5.1}$$

Only inequalities expressed in master variables, such as readiness screening and readiness-cover cuts, are placed directly in (5.1). Workload, energetic, setup-aware, and scheduling-conflict inequalities involve recourse variables and therefore strengthen the component scheduling subproblems rather than the outer master. This separation is important for both validity and interpretation of the computational ablation.

**Compact extensive-form benchmark** The deterministic equivalent of (3.1)–(3.18) provides an independent exact benchmark where its size permits direct solution. The validation uses a restricted single-scenario case for which the extensive form is solved to proven optimality and is matched exactly by C–LBBD. The extensive form is therefore a correctness benchmark, not the claimed scalable solution architecture.

**Exact fixed-readiness preprocessing and scheduling formulations** For a fixed readiness vector and scenario, preprocessing removes only decisions that are provably infeasible. An order is forced to be rejected if at least one required operation has no ready technically eligible machine, or if a relaxed route-processing lower bound already exceeds its release–due window. An operation–machine assignment is removed if fixing that assignment makes the same relaxed route bound exceed the order window. A directed sequencing relation is removed when a relaxed earliest completion of its predecessor, including setup, exceeds a valid latest start of its successor. These operations are validity-preserving reductions; they do not use incumbent objective values and do not remove any feasible accepted schedule.

After preprocessing, a multi-family active component is solved with the exact immediate-successor sequence-dependent setup formulation of Section 3. When only one family remains producible in a component, the implementation uses an exact unordered pairwise disjunctive formulation instead of redundant directed successor arcs. If $k_m$ operations of that family are assigned to machine $m$, the formulation accounts for exactly $k_m - h_{m\omega}$ same-family transitions in the machine load and setup-cost expression. This is an exact reformulation, not a relaxation. The mathematical formulation admits pair-specific big-$M$ constants; the reported computational implementation first performs the exact temporal pruning above and then uses a safe horizon-based constant. Exact subproblems are accepted only when the solver proves optimality; a time-limit incumbent is never substituted for an exact recourse value.

### *5.2. Algorithm 1: exact component-wise logic-based Benders decomposition*

Conditional on $z$, the recourse problem decomposes by technical component and scenario under the conditions of the component-decomposition theorem in Section 4. Each component–scenario problem is solved to global optimality, and its exact value generates the monotone component cut (4.14). The validity of that monotone cut and the stated C–LBBD exactness guarantee are restricted to the off-line

readiness base model, including $\ell_{fm} = 0$. Under the capacity-loss extension, a newly active readiness edge may reduce machine time and recourse value; the same monotone cut is then not justified without a separate validity proof. An exact method for that extension would require appropriately reformulated bounds and cuts, and is not claimed in this paper. Let LB be the best exact feasible first-stage objective obtained from evaluated readiness vectors and let UB be the current globally solved master objective. For numerical implementation, define

$$g_{\text{abs}} = \max\{0, \text{UB} - \text{LB}\}, \qquad g_{\text{rel}} = \frac{g_{\text{abs}}}{\max\{1, |\text{LB}|\}}. \tag{5.2}$$

The exact finite-convergence theorem of Section 4 corresponds to zero tolerances in exact arithmetic. With positive tolerances $\epsilon_{\text{abs}}$ or $\epsilon_{\text{rel}}$, termination provides the associated optimality-gap certificate rather than a literal zero-gap theorem.

**Algorithm 1. Exact component-wise logic-based Benders decomposition (C-LBBD)**

**Require:** Instance data, scenarios $\Omega$, tolerances $\epsilon_{\text{abs}}, \epsilon_{\text{rel}} \geq 0$

**Ensure:** Best feasible readiness vector and certified outer optimality gap; global optimum when both tolerances are zero in exact arithmetic

Initialize LB $\leftarrow -\infty$, incumbent $\leftarrow \varnothing$
Build master (5.1); add master-valid screening/cover inequalities if used
**repeat**
  Solve the outer readiness master to global optimality; let $(z^k, \theta^k)$ be the solution and UB $\leftarrow \text{Obj}^k_{\text{master}}$
  **for all** $c \in C$ and $\omega \in \Omega$ **do**
    Fix readiness to $z^k_c$ and apply exact fixed-readiness preprocessing
    Add any selected recourse-valid workload, energetic, setup, or conflict strengthening
    Solve the exact detailed component OAS recourse problem to proven optimality
  **end for**
  Compute the exact evaluated objective

$$Z^k = -\sum_{f,m} F_{fm} z^k_{fm} + \sum_{\omega \in \Omega} \pi_\omega \sum_{c \in C} Q_c(z^k_c, \omega)$$

  **if** $Z^k >$ LB **then**
    LB $\leftarrow Z^k$; store $z^k$ and its exact recourse schedules as incumbent
  **end if**
  Compute $g_{\text{abs}}$ and $g_{\text{rel}}$ from (5.2)
  **if** $g_{\text{abs}} \leq \epsilon_{\text{abs}}$ or $g_{\text{rel}} \leq \epsilon_{\text{rel}}$ **then**
    **return** incumbent and the certified gap
  **end if**
  **for all** $c \in C$ and $\omega \in \Omega$ **do**
    Add the monotone component optimality cut (4.14) generated at $z^k_c$
  **end for**
**until** termination

When both numerical tolerances are zero, Algorithm 1 is exactly the finite procedure proved in Section 4. With positive tolerances it is a gap-certified implementation of that procedure. The proof requires global solution of every master and component–scenario recourse problem; accordingly, the exact experiments never treat a time-limited subproblem incumbent as an exact value.

**Basic-global ablation reference** To isolate the value of component granularity, the computational study also uses a basic global-cut reference with one expected-recourse variable $\theta$. If $\bar{Q} = \sum_\omega \pi_\omega Q(\bar{z}, \omega)$ and $U = \sum_\omega \pi_\omega \sum_{o \in O_\omega} v_{o\omega}$, its cut is

$$\theta \le \bar{Q} + (U - \bar{Q}) \sum_{e:\bar{z}_e=0} z_e. \tag{5.3}$$

The same monotonicity and gross-value upper-bound argument used for (4.14) establishes validity of (5.3). The basic and component-wise variants use the same exact recourse evaluator; the ablation changes only the granularity of the outer recourse representation and generated monotone cuts.

### *5.3. Algorithm 2: capability-aware fix-and-optimize*

Repeated globally exact stochastic solves can still be expensive. CA–FO therefore searches a restricted neighborhood of the first-stage readiness vector and reoptimizes the associated stochastic recourse. The method is a matheuristic: Its scores guide which readiness edges to release, but the scores themselves are not approximations or bounds on marginal recourse value.

For an incumbent solution $k$, define the expected activity of an active readiness edge $(f, m)$ as

$$A^k_{fm} = \sum_{\omega \in \Omega} \pi_\omega \sum_{\substack{o \in O_\omega:\, f(o)=f \\ r \in \mathcal{R}_o}} p_{orm\omega} x^k_{orm\omega}. \tag{5.4}$$

Let $\kappa_{fm} = 1$ if machine $m$ is technically eligible for at least one operation in the route template of family $f$, and let $\kappa_{fm} = 0$ otherwise. The opportunity score for an inactive edge is

$$O^k_{fm} = \kappa_{fm} \sum_{\omega \in \Omega} \pi_\omega \sum_{o \in O_\omega:\, f(o)=f} (1 - a^k_{o\omega}) v_{o\omega}. \tag{5.5}$$

Thus, low-activity active edges are natural release candidates, whereas high-opportunity inactive edges identify families with rejected value for which the machine offers at least some technical route support. The opportunity score deliberately ignores route complementarity, congestion, setup interactions, and readiness cost; it serves solely as a neighborhood-selection score. Deterministic tie-breaking is used in the implementation.

Let $\mathcal{Z}$ denote the feasible first-stage readiness set defined by (3.2)–(3.3). If $R_k$ is the set of readiness edges released at iteration $k$, define the readiness neighborhood

$$\mathcal{Z}_k(R_k) = \{z \in \mathcal{Z} : z_e = z^k_e \text{ for every } e \notin R_k\}. \tag{5.6}$$

All second-stage decisions are reoptimized for a candidate readiness vector rather than being fixed outside an affected operational region. This is consistent with the implemented validation. In that validation, the released readiness bits are enumerated, budget-infeasible patterns are discarded, and

candidate patterns that are evaluated are solved with exact component recourse; caching avoids duplicate recourse solves. This search therefore provides a feasible improvement mechanism rather than an exact neighborhood certificate. A larger implementation may instead solve the restricted stochastic mixed-integer program or restricted LBBD within a time limit; the acceptance rule below preserves feasibility and monotone incumbent value in either case.

**Algorithm 2. Capability-aware fix-and-optimize (CA-FO)**

**Require:** Feasible incumbent $(z^0, \mathcal{X}^0)$, release sizes $(r^-, r^+)$, optional expanded sizes $(\bar{r}^-, \bar{r}^+)$, stopping rule
**Ensure:** Best feasible solution found
Set $k \leftarrow 0$ and $(z^{\text{best}}, \mathcal{X}^{\text{best}}) \leftarrow (z^0, \mathcal{X}^0)$
**while** the stopping rule is not met **do**
    Compute $A^k_{fm}$ for active edges and $O^k_{fm}$ for inactive edges
    Select up to $r^-$ low-activity active edges and $r^+$ high-opportunity inactive edges; let their union be $R_k$
    Search $\mathcal{Z}_k(R_k)$ and reoptimize recourse for each candidate readiness vector
    **if** a feasible candidate with objective strictly above the incumbent is found **then**
        Accept the best improving candidate and recompute the edge scores
    **else**
        Optionally repeat once with expanded release set $(\bar{r}^-, \bar{r}^+)$
        **if** the expanded neighborhood also gives no feasible improvement **then**
            Stop or move to the next prescribed neighborhood according to the stopping rule
        **end if**
    **end if**
    $k \leftarrow k + 1$
**end while**
**return** $(z^{\text{best}}, \mathcal{X}^{\text{best}})$

The readiness-neighborhood structure is related to the relax-and-fix/fix-and-optimize principle used for intertwined planning and order-acceptance decisions by Aouam et al. [9], but here the released objects are readiness edges and every candidate is valued through the downstream stochastic OAS recourse.

**Proposition 5.1** (Feasibility and monotone incumbent value of CA-FO)**.** *Suppose Algorithm 2 starts from a feasible solution of the original model and accepts only candidate solutions that are feasible for the original model and strictly improve its objective. Then every accepted incumbent is feasible and the incumbent sequence satisfies*

$$Z(\mathcal{X}^{k+1}) \geq Z(\mathcal{X}^k).$$

*If a neighborhood is solved exactly, its best value is also no smaller than the incumbent value because the incumbent readiness vector belongs to* $\mathcal{Z}_k(R_k)$*.*

*Proof.* By definition of (5.6), the incumbent readiness vector satisfies every fixing used to define the current neighborhood and therefore remains first-stage feasible. Its incumbent recourse solution is a feasible completion, so an exact neighborhood optimization cannot return a value below the incumbent value. Under a time-limited neighborhood solve, the algorithm accepts only a candidate that is feasible

for the original model and has a strictly larger objective. Otherwise it retains the incumbent. Hence feasibility is preserved and the accepted incumbent objective is nondecreasing. □

CA–FO has no global-optimality guarantee unless the released readiness set eventually spans the full first-stage domain and the resulting full stochastic problem is solved to global optimality. The computational validation therefore makes a narrower claim: On one instance with an independently known exact optimum, the same deterministic search rule is started from four pre-specified readiness vectors. The experiment verifies feasibility preservation, monotone nontrivial improvement from nonoptimal starts, and exact recovery on that validation instance without implying global optimality in general.

## 6. Computational study and application scope

### *6.1. Experimental design and scenario generation*

The synthetic benchmark varies instance size, technical flexibility, readiness scarcity, setup burden, order-regime dependence, and independent replication. These factors define a $3\times3\times2\times2\times2\times10 = 720$-cell design space. The reported structural analyses use a balanced 48-instance subset spanning the factor levels, while the stochastic policy study uses the selected correlated-lumpy instance described below. Training scenarios determine readiness decisions and independently generated scenarios are used for out-of-sample (OOS) evaluation. Table 3 summarizes the experimental design.

**Table 3.** Synthetic HMLV benchmark and computational validation design.

| Block | Factor | Levels / setting | Role |
|---|---|---|---|
| Primary factorial | Instance size | small / medium / large | 3 levels |
| Primary factorial | Technical flexibility | low / medium / high | 3 levels |
| Primary factorial | Readiness scarcity | tight / loose | $\chi = 0.35/0.60$ |
| Primary factorial | Setup burden | low / high | $\psi = 0.15/0.45$ |
| Primary factorial | Order regime | independent-intermittent / correlated-lumpy | 2 levels |
| Design space | Independent replications | 10 per factor cell | 720-cell generator design space |
| Structural subset | Balanced instances | 48 | spans factor levels for structural analysis |
| SAA | Training scenarios | $N \in \{6, 12, 24, 48\}$ | 5 independent replications per $N$ |
| SAA stability | Common OOS scenarios | 200 | independently generated common OOS bank |
| Main policy study | Primary training size | 24 | selected after the stability analysis |
| Main policy study | Selected training bank | first 24-scenario replication | same readiness network as all SAA replications |
| Information value | Wait-and-see enumeration | 24 scenarios × 46 feasible readiness patterns | exact |

**Operational elicitation of cost and capacity coefficients** For a deployment, the first-stage readiness cost $F_{fm}$ can be estimated from documented fixture/tooling maintenance, program validation, qualification renewal, and planned engineering or technician hours multiplied by relevant labor rates. The corresponding resource $b_{fm}$ and budget $B$ must be measured in the same unit (e.g., engineering-hours during the preparation window, or monetized readiness-budget units); double counting expenditures in both $F_{fm}$ and $b_{fm}$ should be avoided. Operation-specific processing times $p_{orm\omega}$ and setup times $s_{ff'm}$ may be estimated using machine/MES execution logs after checking family, machine, and route identifiers, while order values should be based on incremental contribution margins rather than gross

sales. ERP order histories can inform intermittent arrival and size distributions, with customer- or project-level grouping used to evaluate cross-order dependence. These are proposed measurement procedures, not claims that the synthetic benchmark coefficients were fitted to plant records.

Figure 1 confirms that technical flexibility, readiness scarcity, and setup burden vary as distinct mechanisms rather than as a single size effect.

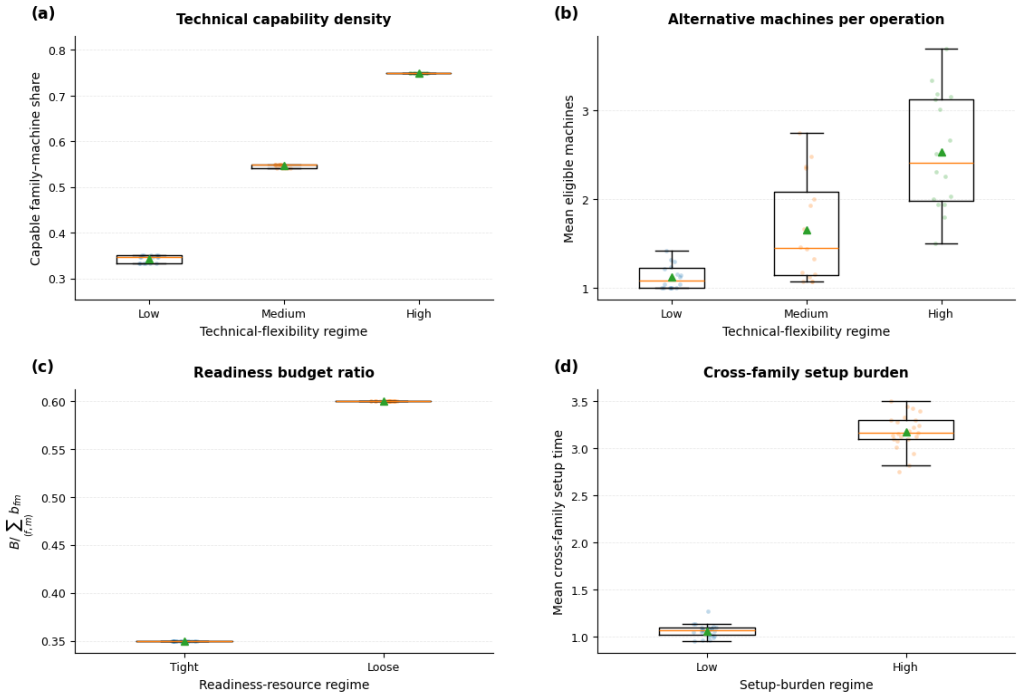


**Figure 1.** HMLV benchmark and readiness-network structure. Panels report (a) technical capability density, (b) mean alternative machines per operation, (c) readiness-budget ratio, and (d) cross-family setup burden across the structural benchmark subset.

The scenario generator separates order occurrence from positive order size and includes a correlated-lumpy regime in which a common state affects arrivals, offered value, and due-date pressure. Scenario probabilities are used throughout. Figure 2 shows the resulting heterogeneity and the imperfect coupling between offered value and route-lower-bound workload.

### *6.2. SAA stability and training-size choice*

For the selected correlated-lumpy instance, the SAA stability analysis used $N \in \{6, 12, 24, 48\}$ with five independent replications per $N$. All 20 optimization runs selected the same readiness network, $\{\mathrm{F002}-\mathrm{M003}, \mathrm{F004}-\mathrm{M001}\}$, so pairwise Jaccard similarity and modal share are one at every tested size. Because the network was already stable at $N = 12$ in this focal instance, we use $N = 24$ for the original computational illustration. This is not a universal scenario-size recommendation: The cross-instance feasible-heuristic analysis below cannot establish stability of exact SAA optima. Increasing $N$ to 48 leaves both the selected network and common-OOS value unchanged in this focal experiment, while mean solution time rises from 63.95 s to 248.89 s.

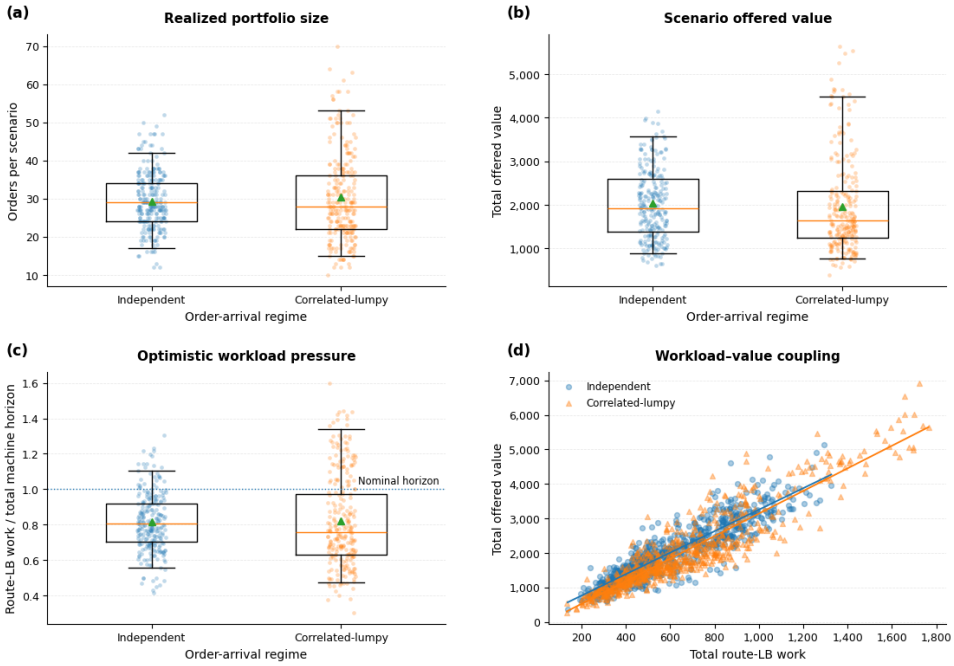


**Figure 2.** Stochastic HMLV order environment and demand heterogeneity. Panels show (a) realized portfolio size, (b) total offered value, (c) optimistic route-lower-bound workload pressure, and (d) workload–value coupling for independent-intermittent and correlated-lumpy scenarios.

**Cross-instance heuristic sensitivity to SAA training size** As a separate diagnostic, we evaluated three independently generated training scenario banks at each of $N = 24$ and $N = 48$ for each of 48 prespecified structural instances, totaling 288 feasible-heuristic training runs. The mean within-size readiness-network Jaccard similarities were 0.8079 ($N = 24$) and 0.8149 ($N = 48$), and the mean cross-size similarity was 0.8100. Across instances, the mean relative change in the heuristic OOS objective in favor of $N = 48$ was 1.256%. These results describe *heuristic* network consistency, not stability of exact SAA optimal solutions, and do not imply that $N = 24$ should be a universal default.

### *6.3. Policy value and value of information*

Table 4 compares all policies on the same 200 independently generated OOS scenarios. The stochastic, aggregate schedule-ignorant, and value-ranked rules select the same two-edge network and therefore attain the same mean net objective, 145.89, with a 95% confidence interval of [132.60, 159.18]. This tie shows that the value of schedule-aware readiness design is instance-dependent and arises when detailed scheduling changes the preferred readiness network. The deterministic-medoid, workload-ranked, and volume-ranked rules select the alternative two-edge network and obtain 65.29, despite a higher acceptance rate (0.235 versus 0.149), consistent with the value–volume reversal in Section 4.

**Table 4.** Main policy comparison and information-value measures. All OOS policies are evaluated on the same 200 scenarios.

**Panel A. Out-of-sample policy performance**

| Policy | Ready edges | Readiness cost | OOS net objective (95% confidence interval) | Acceptance rate | Accepted value | Loss vs. proposed (%) |
|---|---|---|---|---|---|---|
| Proposed stochastic | 2 | 24.11 | 145.89 [132.60, 159.18] | 0.149 | 222.21 | 0.00 |
| Aggregate / schedule-ignorant | 2 | 24.11 | 145.89 [132.60, 159.18] | 0.149 | 222.21 | 0.00 |
| Deterministic medoid | 2 | 22.72 | 65.29 [60.52, 70.06] | 0.235 | 199.28 | 55.25 |
| Workload-ranked | 2 | 22.72 | 65.29 [60.52, 70.06] | 0.235 | 199.28 | 55.25 |
| Value-ranked | 2 | 24.11 | 145.89 [132.60, 159.18] | 0.149 | 222.21 | 0.00 |
| Volume-ranked | 2 | 22.72 | 65.29 [60.52, 70.06] | 0.235 | 199.28 | 55.25 |
| No readiness | 0 | 0.00 | 0.00 [0.00, 0.00] | 0.000 | 0.00 | 100.00 |

**Panel B. Information-value measures on the 24-scenario training distribution**

| Metric | Value | Interpretation |
|---|---|---|
| Recourse problem (RP) | 144.172 | Stochastic-program objective on the 24-scenario training bank. |
| Wait-and-see (WS) | 160.224 | Probability-weighted wait-and-see objective with scenario-specific readiness. |
| Expected value of perfect information (EVPI) | 16.053 | $WS - RP$ for the maximization model. |
| Deterministic-medoid expected value | 65.167 | Expected training-distribution value of the deterministic-medoid readiness. |
| Stochastic-design gain | 79.005 | Difference between RP and the deterministic-medoid expected value. |

For the 24-scenario training distribution, $RP = 144.172$ and $WS = 160.224$, yielding $EVPI = 16.053$. The deterministic-medoid network has expected value 65.167, giving a stochastic-design gain of 79.005. The classical value of the stochastic solution (VSS) is not reported because a componentwise expected job set is not a valid realization of the random finite order portfolio. Figure 3 provides paired OOS evidence: The mean gain over the deterministic medoid is 80.60, with a 95% confidence interval of $[68.32, 92.88]$, and is positive in 170 of 200 scenarios.

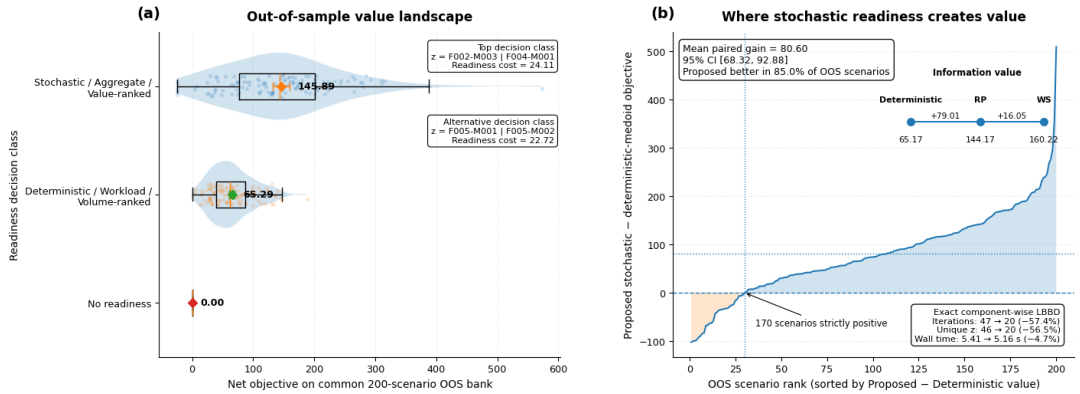


**Figure 3.** Policy value, information value, and exact-search evidence on the selected correlated-lumpy instance. (a) OOS value distributions for the three unique readiness-decision classes on the common 200-scenario bank. (b) Ranked paired gain of the stochastic readiness decision over the deterministic-medoid decision; the inset reports the deterministic-design gain and EVPI, and the lower annotation summarizes the exact LBBD ablation.

### *6.4. Cross-instance common-pool policy comparison*

The original exact-policy comparison in Table 4 uses one focal instance and a common bank of 200 OOS scenarios; there, the stochastic, aggregate, and value-ranked decisions choose the same readiness network and therefore tie. To test whether this outcome is representative of the broader benchmark, we evaluated 48 prespecified structural instances (16 per size) using a *separate, controlled finite-candidate-pool experiment*. For each instance, a common set of feasible first-stage readiness networks was constructed from training information, pre-existing policy choices, and predefined local alterations *without consulting OOS outcomes*. All six selection rules then chose from exactly the same candidate set under identical readiness feasibility and resource-budget restrictions: the schedule-aware training heuristic, an approximate aggregate-capacity training rule, and deterministic representative-scenario heuristics based on medoid, mean, high-workload P90, and low-workload P10 scenarios. Each rule was selected using the same 24-scenario first-replication training bank (the deterministic rules use one training representative with probability one) and assessed on the same independent 100-scenario OOS bank within each instance. The shared pool eliminates differences in *candidate access*, but it does not make these methods exact: Their training evaluation functions differ and OOS recourse is computed by a common feasible scheduling heuristic. Neither the exact Table 4 policies nor full optimal SAA solutions are claimed to have been recomputed on the 48 instances.

For comparator $b$ in instance $i$, define the paired OOS performance difference as $D_{i,b} = \widehat{V}_i^{\mathrm{SA}} - \widehat{V}_i^{b}$; a positive value favors the schedule-aware heuristic. Each instance, not each individual policy or scenario,

contributes one paired observation. We report the sample mean over 48 instances, a two-sided Student-$t$ 95% confidence interval, Cohen's paired effect size $d_z = \overline{D}/s_D$, and Holm-adjusted two-sided paired-$t$ $p$ values across five comparisons. These statistics describe the finite prespecified synthetic benchmark; they must not be interpreted as industry-wide population inference or as identification of a causal schedule-awareness effect.

**Table 5.** Controlled common-candidate-pool feasible-heuristic policy comparison over 48 structural instances. All differences are mean OOS net-objective units (schedule-aware minus comparator) on 100 common OOS scenarios per instance, *evaluated by the same feasible scheduling heuristic*; positive values favor the schedule-aware rule under that evaluator. B/T/W denote the number of instances with better/tied/worse performance. These are not exact Table 4 optimizations, and policy-specific evaluator gaps are not controlled.

| Comparator | Mean difference | 95% paired $t$ CI | $d_z$ | B/T/W | Holm-adjusted $p$ |
|---|---|---|---|---|---|
| Aggregate-capacity proxy | +28.99 | [+14.89, +43.09] | 0.597 | 30/15/3 | 0.000249 |
| Deterministic medoid | +27.78 | [+16.92, +38.65] | 0.743 | 28/17/3 | 0.0000257 |
| Deterministic mean | +28.82 | [+16.38, +41.26] | 0.673 | 25/19/4 | 0.0000790 |
| Deterministic high workload (P90) | +31.81 | [+16.51, +47.11] | 0.604 | 27/20/1 | 0.000249 |
| Deterministic low workload (P10) | +44.97 | [+27.09, +62.85] | 0.730 | 32/12/4 | 0.0000275 |

Table 5 shows a mean advantage of 28.99 objective units over the aggregate proxy (95% paired interval [14.89,43.09], Holm-adjusted $p = 0.000249$), with 30 positive, 15 tied, and three negative instance comparisons. The largest positive aggregate-proxy difference was +287.24 on M_FH_RL_SL_COR_R10. The mean difference remains +23.49 when this single large positive case is excluded (47 remaining instances; descriptive sensitivity check, not a re-estimated primary test). The three unfavorable comparisons were M_FM_RL_SH_IND_R06 (−17.13), L_FM_RT_SH_COR_R08 (−29.85), and L_FH_RT_SH_IND_R08 (−9.67). All 48 cases, including these exceptions and all ties, were retained. The earlier broad nine-policy exploratory screening remains a separate heuristic audit and is not pooled into the common-pool inference reported here. Importantly, evaluating different readiness patterns by the same feasible scheduling procedure does not ensure equal suboptimality gaps across policies. The limited matched-scenario exact-recourse verification below demonstrates that unequal evaluator gaps can affect even the sign of a pairwise comparison. Consequently, the 48-instance means and their associated paired tests quantify *heuristic-evaluated* policy differences, not exact-recourse population advantages.

Figure 4 displays every instance-level difference together with the five paired confidence intervals. The outlying positive case is labeled explicitly rather than silently omitted; the limits of the feasible-heuristic evaluator stated above also apply to this visualization.

Table 6 shows variation across the benchmark: Mean differences were +16.78, +38.79, and +31.39 for small, medium, and large instances, respectively. Larger values were observed under high technical flexibility (+49.61) than under low flexibility (+18.40), but no statistical test establishes that the difference *between factor levels* is significant. In particular, the high-setup group did not show a consistently larger schedule-aware advantage than the low-setup group. These are conditional descriptions rather than universal managerial rules.

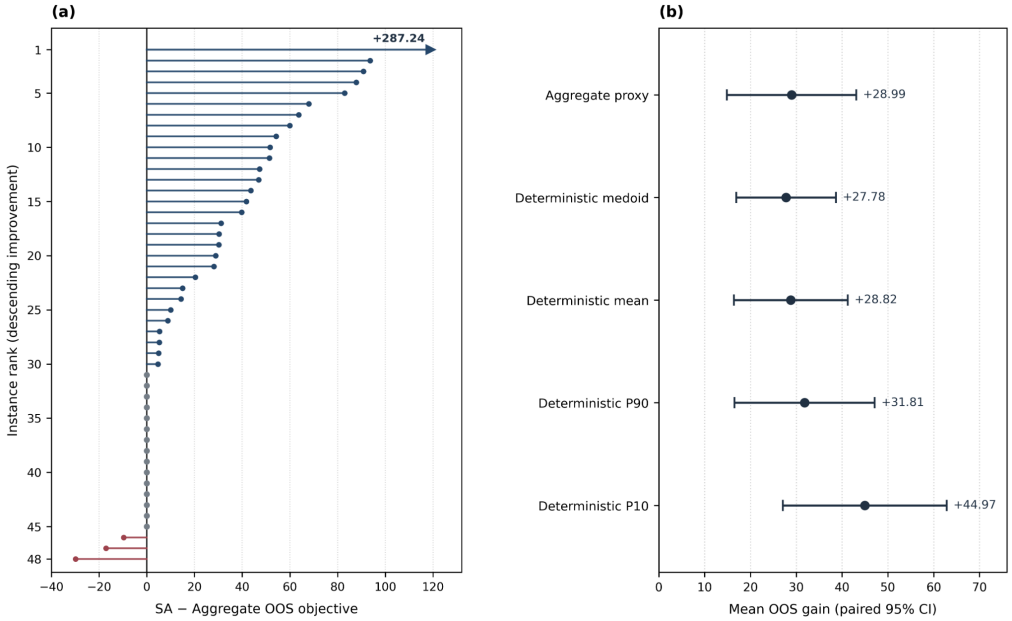


**Figure 4.** Cross-instance comparisons of feasible-heuristic policy performance on 48 prespecified synthetic HMLV instances. (a) Sorted paired OOS net-objective differences between schedule-aware and aggregate-capacity policies (30 positive, 15 tied, and three negative instances); the extreme +287.24 difference lies beyond the displayed horizontal-axis range and is indicated by an arrow. (b) Mean paired differences and 95% confidence intervals against five reference policies. All first-stage choices are taken from a common finite candidate pool and OOS values are evaluated by a common feasible scheduling heuristic; neither panel demonstrates exact first-stage or exact 48-instance recourse optimality.

**Table 6.** Descriptive structural heterogeneity in the *common-pool* schedule-aware minus aggregate-proxy OOS difference. Units are net-objective units, not percentages; structural-factor levels describe the chosen benchmark and do not imply causal effects.

| Factor | Level | Instances | Mean difference | B/T/W |
|---|---|---|---|---|
| Size | Small | 16 | +16.78 | 9/7/0 |
| | Medium | 16 | +38.79 | 10/5/1 |
| | Large | 16 | +31.39 | 11/3/2 |
| Technical flexibility | Low | 16 | +18.40 | 9/7/0 |
| | Medium | 16 | +18.96 | 9/5/2 |
| | High | 16 | +49.61 | 12/3/1 |
| Readiness scarcity | Tight | 23 | +19.39 | 10/11/2 |
| | Loose | 25 | +37.82 | 20/4/1 |
| Setup burden | High | 24 | +21.11 | 15/6/3 |
| | Low | 24 | +36.87 | 15/9/0 |
| Order regime | Correlated-lumpy | 24 | +37.97 | 17/6/1 |
| | Independent-intermittent | 24 | +20.01 | 13/9/2 |

An additional exploratory comparison of the five predeclared structural factors was conducted with 10,000 fixed-seed label permutations on the 48 paired instance-level differences. The unadjusted permutation $p$ values were 0.4885 (size), 0.0905 (technical flexibility), 0.2028 (readiness scarcity), 0.2929 (setup burden), and 0.2256 (order regime); the smallest Holm-adjusted value was 0.4525. None

of these exploratory factor-level contrasts was significant at 5%. We therefore do not interpret observed differences between strata as statistically established or causal; furthermore, the synthetic structural instances do not constitute a randomized industrial population. Reproducible tests and confidence calculations are retained in the research audit trail.

**Readiness-cost and budget sensitivity** We also performed a 3×3 finite-pool sensitivity analysis for each of the same 48 structural instances. Readiness fixed costs were multiplied by 0.75, 1.00, or 1.25, and available readiness resource budgets were multiplied by 0.75, 0.90, or 1.00. For each combination, a training-score maximizer was reselected from the *same predeclared feasible candidate pool* subject to the adjusted budget; the OOS net value was then evaluated using the common feasible scheduler and the adjusted fixed-cost term. Table 7 reports the resulting 3×3 sensitivity grid across the 48 structural instances. Thus, all 432 values below are model-based perturbations of synthetic coefficients, not observations or plant-calibrated uncertainty ranges.

**Table 7.** Readiness-cost and resource-budget sensitivity: average OOS net objective across 48 structural instances. Cost multipliers alter first-stage fixed costs and budget multipliers alter admissible readiness resources; values arise from reselection within the common finite candidate pools, not exact reoptimization or plant-specific cost estimation.

| Readiness resource budget | Fixed cost ×0.75 | Fixed cost ×1.00 | Fixed cost ×1.25 |
|---|---|---|---|
| 0.75$B$ | 276.24 | 251.72 | 230.46 |
| 0.90$B$ | 396.06 | 356.02 | 319.20 |
| 1.00$B$ | 398.25 | 357.15 | 321.01 |

In this benchmark, the mean OOS value rises considerably as the budget multiplier increases from 0.75 to 0.90; the additional change from 0.90 to 1.00 is small. This is evidence of a *finite-pool plateau under the specified perturbations*, not of a general threshold or an estimated industrial budget elasticity. Several budget-tight settings select the no-readiness fallback and yield zero OOS value; their inclusion is intentional. In application, readiness fixed costs and budget resources must be estimated from actual engineering and production records as described in the operational parameter-elicitation paragraph of this section; the present synthetic sensitivity exercise cannot itself validate those estimates.

### *6.5. Limited matched-scenario exact fixed-readiness verification*

To check whether the common feasible scheduling evaluator changes the policy ranking, we fixed the two previously selected readiness networks in each of three prespecified small instances (one original tie and two positive heuristic comparisons). Without inspecting their values, eight of each instance's 100 OOS scenario IDs were selected by a fixed hash rule. On these *same eight scenarios*, both readiness patterns were evaluated using (i) the original feasible scheduling heuristic and (ii) the exact Step10B fixed-readiness mixed-integer recourse model. All 80 component-policy tasks were certified optimal by Gurobi; repeated identical patterns explain why 64 unique solution checkpoints suffice. All 48 policy–scenario pairs obeyed the valid bound that exact fixed-readiness recourse is at least as good as the feasible heuristic under the same model.

**Table 8.** Limited fixed-readiness exact recourse cross-check. Entries are schedule-aware minus aggregate-proxy net-objective differences. The 8-scenario heuristic and exact columns use identical preselected OOS scenario IDs and fixed first-stage networks; the 100-scenario heuristic column is a different sample and is shown solely as a reference. The last two columns decompose the difference between Exact(8) and Heuristic(100), not a causal effect.

| Small instance | Heuristic(100) | Heuristic(8) | Exact(8) | Sample contrast | Evaluator contrast |
|---|---|---|---|---|---|
| S_FL_RT_SL_COR_R02 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| S_FL_RL_SL_COR_R08 | +30.31 | +43.46 | −9.29 | +13.15 | −52.75 |
| S_FH_RT_SH_IND_R09 | +51.44 | +69.01 | +29.26 | +17.57 | −39.75 |

In Table 8, the first small instance selects identical networks under both policies, and both evaluators tie. For S_FL_RL_SL_COR_R08, the 100-scenario heuristic advantage of +30.31 becomes +43.46 under the same eight scenarios, but the fixed-readiness exact value favors the aggregate proxy by 9.29. The +13.15 change is attributable arithmetically to restricting the heuristic to eight scenarios, whereas the −52.75 change arises from replacing the feasible heuristic by exact recourse on *the same eight scenarios*; the aggregate network benefited more from exact recourse in this sample. For S_FH_RT_SH_IND_R09, both evaluators favor the schedule-aware network, although the exact difference (+29.26) is smaller than the same-eight heuristic difference (+69.01). This mixed evidence is a methodological limitation of the broader 48-instance feasible-heuristic comparisons: It does not invalidate their reported surrogate-evaluator statistics, but it prevents interpreting them as certified exact-policy outperformance. The check covers only three selected small instances and 8/100 OOS scenarios per instance; it neither certifies the 100-scenario population means nor reoptimizes first-stage SAA decisions. We report all three outcomes, including the reversal, without extrapolating their frequencies to the full benchmark.

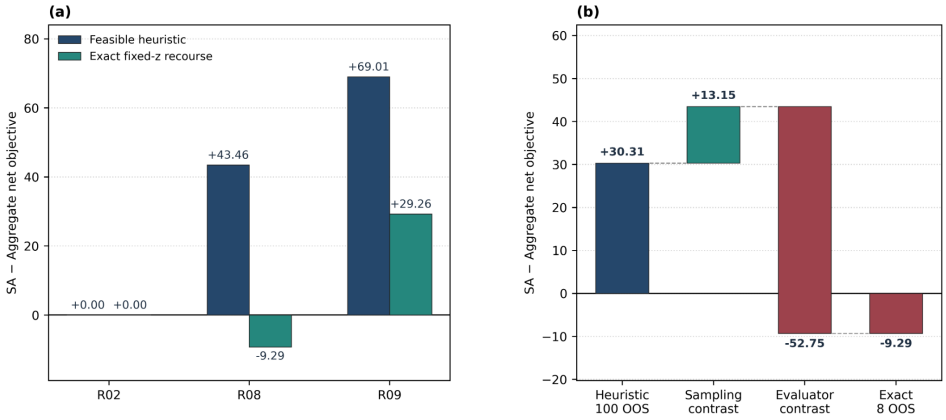


**Figure 5.** Matched-scenario evaluation of fixed first-stage readiness networks on three small instances. (a) Schedule-aware-minus-aggregate net-objective differences computed with the common feasible scheduler and with exact mixed-integer recourse, using the same eight hash-selected OOS scenarios within each instance. (b) Arithmetic decomposition of the ranking reversal in instance S_FL_RL_SL_COR_R08: The original 100-OOS heuristic difference (+30.31), the eight-scenario sampling contrast (+13.15), and the exact-versus-heuristic evaluator contrast (−52.75) yield an eight-scenario exact difference of −9.29. These are limited fixed-readiness recourse checks, not exact evaluation of all 100 OOS scenarios or global reoptimization of the first-stage SAA problem.

Figure 5 visualizes the matched-scenario comparison and the arithmetic decomposition of the R08 reversal. It includes the unfavorable exact outcome rather than presenting the heuristic advantage as a certified exact-policy gain.

### *6.6. Exactness, decomposition strength, and scalability*

Table 9 consolidates exactness and scalability evidence. C–LBBD matches the independent extensive-form optimum 325.376873, while CA–FO reaches the known optimum 370.473687 from four deterministic starts, demonstrating nontrivial monotone improvement on that validation instance. CA–FO remains a heuristic outside this validation setting. The two exact reference values concern the same frozen pilot instance, S_FL_RT_SL_IND_R05, but *different training-scenario sets*. The objective 325.376873 is the single-scenario (TR_001, probability one) extensive-form and C–LBBD validation, while 370.473687 is the C–LBBD optimum of the six-scenario training formulation, which is also recovered by CA–FO in the separate multi-start exercise. These are distinct optimization problems, not conflicting optima of the same input bank. They must not be used to form a relative optimality gap. The original experiment notebook records these checks in Steps 06A, 07, and 09C. In the LBBD ablation, both variants attain 81.726942, while component-wise cuts reduce master iterations from 47 to 20 and unique readiness vectors from 46 to 20. Across all SAA banks, C–LBBD satisfies the prescribed optimality tolerance for $N = 6, 12, 24, 48$. Temporal preprocessing together with the exact single-family reformulation reduces the number of sequencing binaries by 65.5–88.9% relative to the raw directed-pair count for the tested active patterns.

### *6.7. Interpretation and scope*

The results show that the value of stochastic readiness design depends primarily on which product–machine relations are maintained rather than on the number of active links alone. On the selected correlated-lumpy instance, the stochastic model substantially outperforms the deterministic-medoid design. By contrast, the aggregate schedule-ignorant and value-ranked policies attain the same OOS value when they select the same readiness network. The computational evidence therefore supports a conditional interpretation: Stochastic modeling creates value when uncertainty changes the preferred composition of the readiness network, not merely because a stochastic model is used.

The framework is most relevant to HMLV settings in which readiness must be committed before the future order portfolio is known, order acceptance remains discretionary, and operational feasibility depends on detailed alternative-machine assignment and sequencing. Representative civilian applications include customized industrial production, aerospace replacement components, rail and energy-service parts, and legacy-equipment support. Defense sustainment provides an additional application setting [48, 49], but the mathematical formulation is sector independent.

Because the numerical study uses a synthetic benchmark, the reported objective values should not be interpreted as estimates for a particular industrial system. The experiments are designed instead to isolate the effects of readiness scarcity, demand uncertainty, value heterogeneity, and detailed scheduling. Validation on sector-calibrated industrial data is an important direction for subsequent research.

**Table 9.** Algorithm validation, decomposition ablation, exact scalability, and formulation compaction.

**Panel A. Exactness and CA–FO validation**

| Evidence | Reference objective | Compared objective | Absolute gap | Relative gap | Result |
|---|---|---|---|---|---|
| Extensive form vs. C–LBBD | 325.376873 | 325.376873 | 0.00e+00 | – | Exact agreement |
| CA–FO (zero-readiness start) | 370.473687 | 370.473687 | 1.14e-13 | 3.07e-16 | Exact match |
| CA–FO (cheapest-edge start) | 370.473687 | 370.473687 | 1.14e-13 | 3.07e-16 | Exact match |
| CA–FO (volume-ranked start) | 370.473687 | 370.473687 | 1.14e-13 | 3.07e-16 | Exact match |
| CA–FO (value-ranked start) | 370.473687 | 370.473687 | 1.14e-13 | 3.07e-16 | Exact match |

**Panel B. Basic-global versus component-wise exact LBBD**

| Variant | Objective | Iterations | Cuts | Unique $z$ evaluated | Wall time (s) | Final relative gap |
|---|---|---|---|---|---|---|
| Basic global | 81.726942 | 47 | 46 | 46 | 5.407 | 1.74e-15 |
| Component-wise | 81.726942 | 20 | 228 | 20 | 5.156 | 1.27e-14 |

**Panel C. Exact C–LBBD scaling over SAA training size**

| $N$ | Replications | Mean runtime (s) | Runtime standard deviation (s) | Mean iterations | Optimal solved (%) |
|---|---|---|---|---|---|
| 6 | 5 | 10.41 | 12.50 | 20.0 | 100.0 |
| 12 | 5 | 40.04 | 38.52 | 20.0 | 100.0 |
| 24 | 5 | 63.95 | 48.41 | 20.0 | 100.0 |
| 48 | 5 | 248.89 | 234.68 | 20.0 | 100.0 |

**Panel D. Exact sequencing-variable compaction**

| Readiness pattern | OOS scenarios | Mean raw directed arcs | Mean retained sequencing binaries | Reduction (%) |
|---|---|---|---|---|
| F005-M001\|F005-M002 | 200 | 393.8 | 135.8 | 65.5 |
| F002-M003\|F004-M001 | 200 | 33.5 | 3.7 | 88.9 |
| No readiness | 200 | 0.0 | 0.0 | 0.0 |

## 7. Conclusion

This paper develops a two-stage stochastic HMLV production-readiness problem in which family–machine readiness is committed before an uncertain order portfolio is observed, after which orders are accepted or rejected, assigned to ready machines, and scheduled with sequence-dependent setups and hard delivery constraints. The formulation links a sparse pre-demand readiness network directly to operation-level OAS recourse. The structural analysis establishes strong NP-hardness, the boundary of readiness monotonicity, non-submodularity, aggregate-capacity insufficiency, value–volume reversal, and technical-component decomposition, which in turn support readiness-cover, workload, energetic, conflict, and component-wise LBBD cuts. Computationally, the selected readiness pattern is stable across 20 SAA replications; on the common 200-scenario OOS bank it attains mean net value 145.89 versus 65.29 for the deterministic-medoid design. The aggregate and value-ranked baselines attain the same value because they select the same readiness network. C–LBBD matches an independent extensive-form optimum, and the ablation reduces master iterations from 47 to 20 relative to the basic global-cut variant, although the wall-clock improvement on this validation instance is modest. CA–FO also exhibits monotone, nontrivial improvement on a validation case with a known optimum.

A separate 48-instance common-finite-pool heuristic comparison yields an average OOS net-objective advantage of 28.99 relative to an aggregate-capacity proxy and positive average differences against four representative-scenario heuristic controls. The three negative and 15 tied aggregate comparisons show that this advantage is not universal. Structural contrasts and a 3×3 cost–budget sensitivity grid provide conditional operational diagnostics. These results are based on feasible-heuristic OOS recourse, not certified exact-recourse rankings. A limited three-instance matched-scenario pilot certifies fixed-readiness exact recourse and yields one tie, one reversal in favor of the aggregate proxy, and one schedule-aware advantage; consequently, the heuristic-evaluated differences cannot be taken as general exact-policy superiority. Structural-factor subgroup differences also fail exploratory multiplicity-adjusted significance tests. The original focal instance's exact-policy tie remains valid.

The study is limited by its use of synthetic instances and therefore supports structural conclusions rather than sector-specific performance estimates. In addition, the main policy comparison is centered on the selected correlated-lumpy instance, so broader empirical generalization requires further validation across calibrated operating environments. Future work should therefore examine sector-calibrated industrial data, wider multi-regime computational studies, multi-period readiness adaptation, readiness degradation, and extensions in which qualification or readiness activities consume endogenous production capacity. These directions would test the robustness of the present structural results while extending the framework toward longer-horizon and operationally calibrated HMLV planning settings.

## Use of Generative-AI tools declaration

Generative artificial intelligence (AI) tools were used only for limited language polishing and grammatical refinement. The authors independently developed the research design, mathematical formulation, data analysis, computational experiments, interpretation of results, and final manuscript content. The authors reviewed all AI-assisted edits and take full responsibility for the published work.

## Author contributions

Youngchul Kim: Conceptualization, methodology, mathematical formulation, theoretical analysis, algorithm design, computational experimentation, validation, visualization, investigation, and writing–original draft. Chaehwa Lee: Conceptualization, methodology, operational validation, military logistics and sustainment interpretation, assessment of field applicability, and writing–review and editing. Hyeyoung Hwang: Investigation, operational contextualization, validation of practical assumptions, assessment of field relevance, interpretation of computational results, and writing–review and editing. Jinho Cha: Methodology, mathematical analysis, algorithm verification, computational validation, reproducibility auditing, interpretation of numerical results, visualization, and writing–review and editing. Hongsuk Park: Supervision, conceptualization, methodology, validation, research oversight, critical review of mathematical and managerial implications, and writing–review and editing. All authors discussed the research design, reviewed the mathematical formulation and computational findings, contributed to the interpretation of the results, critically revised the manuscript, and approved the final version. Hongsuk Park serves as the corresponding author.

## Acknowledgments

This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP)–ITRC (Information Technology Research Center) grant funded by the Korean government (MSIT) (IITP-2026-RS-2024-00438430, 90%) and by Kumoh National Institute of Technology (2026) through the Gyeongbuk ANCHOR system–(Defense R&D Spin-On) program of the Gyeongbuk ANCHOR Center, funded by the Ministry of Education (MOE) and Gyeongsangbuk-do, Republic of Korea (2026-anchor-15-105, 10%).

## Conflict of interest

The authors declare that they have no competing interests related to this work.

## References

1. S. Dauzère-Pérès, J. Ding, L. Shen, K. Tamssaouet, The Flexible Job Shop Scheduling Problem: A Review, *Eur. J. Oper. Res.*, **314** (2024), 409–432. https://doi.org/10.1016/j.ejor.2023.05.017
2. C. Özgüven, L. Özbakır, Y. Yavuz, Mathematical Models for Job-Shop Scheduling Problems with Routing and Process Plan Flexibility, *Appl. Math. Model.*, **34** (2010), 1539–1548. https://doi.org/10.1016/j.apm.2009.09.002
3. S. A. Slotnick, Order Acceptance and Scheduling: A Taxonomy and Review, *Eur. J. Oper. Res.*, **212** (2011), 1–11. https://doi.org/10.1016/j.ejor.2010.09.042
4. L. Xu, Q. Wang, S. Huang, Dynamic Order Acceptance and Scheduling Problem with Sequence-Dependent Setup Time, *Int. J. Prod. Res.*, **53** (2015), 5797–5808. https://doi.org/10.1080/00207543.2015.1005768

5. Q. Li, D. Zhang, S. Wang, I. Kucukkoc, A Dynamic Order Acceptance and Scheduling Approach for Additive Manufacturing On-Demand Production, *Int. J. Adv. Manuf. Technol.*, **105** (2019), 3711–3729. https://doi.org/10.1007/s00170-019-03796-x
6. M. Fu, R. G. Askin, J. W. Fowler, M. Zhang, Stochastic Optimization of Product–Machine Qualification in a Semiconductor Back-End Facility, *IIE Trans.*, **47** (2015), 739–750. https://doi.org/10.1080/0740817X.2014.964887
7. X. Chang, M. Dong, Stochastic Programming for Qualification Management of Parallel Machines in Semiconductor Manufacturing, *Comput. Oper. Res.*, **79** (2017), 49–59. https://doi.org/10.1016/j.cor.2016.09.006
8. H. Li, W. T. Huh, M. C. Sampaio, N. Keng, Planning Production and Equipment Qualification under High Process Flexibility, *Prod. Oper. Manag.*, **30** (2021), 3369–3390. https://doi.org/10.1111/poms.13439
9. T. Aouam, K. Geryl, K. Kumar, N. Brahimi, Production Planning with Order Acceptance and Demand Uncertainty, *Comput. Oper. Res.*, **91** (2018), 145–159. https://doi.org/10.1016/j.cor.2017.11.013
10. F. Xiong, L. Jing, C. Xiang, Y. Han, Order Acceptance and Scheduling under Eligibility, Availability, and Budget Constraints in Distributed Heterogeneous Flow Shop Production, *Expert Syst. Appl.*, **280** (2025), 127586. https://doi.org/10.1016/j.eswa.2025.127586
11. F. Tanrisever, D. J. Morrice, D. P. Morton, Managing Capacity Flexibility in Make-to-Order Production Environments, *Eur. J. Oper. Res.*, **216** (2012), 334–345. https://doi.org/10.1016/j.ejor.2011.07.043
12. W. C. Jordan, S. C. Graves, Principles on the Benefits of Manufacturing Process Flexibility, *Manag. Sci.*, **41** (1995), 577–594. https://doi.org/10.1287/mnsc.41.4.577
13. D. Simchi-Levi, Y. Wei, Understanding the Performance of the Long Chain and Sparse Designs in Process Flexibility, *Oper. Res.*, **60** (2012), 1125–1141. https://doi.org/10.1287/opre.1120.1081
14. G. Zhang, J. Sun, X. Liu, G. Wang, Y. Yang, Solving Flexible Job Shop Scheduling Problems with Transportation Time Based on Improved Genetic Algorithm, *Math. Biosci. Eng.*, **16** (2019), 1334–1347. https://doi.org/10.3934/mbe.2019065
15. S. Yan, G. Zhang, J. Sun, W. Zhang, An Improved Ant Colony Optimization for Solving the Flexible Job Shop Scheduling Problem with Multiple Time Constraints, *Math. Biosci. Eng.*, **20** (2023), 7519–7547. https://doi.org/10.3934/mbe.2023325
16. Y. Chen, H. Wang, C. Tsai, C. Wu, A Genetic Algorithm for Tool-Constrained Scheduling with Dynamic Expedited Orders in Smart Manufacturing, *J. Ind. Manag. Optim.*, **22** (2026), 182–213. https://doi.org/10.3934/jimo.2026007
17. J. C. Chen, C. Wu, C. Chen, K. Chen, Flexible Job Shop Scheduling with Parallel Machines Using Genetic Algorithm and Grouping Genetic Algorithm, *Expert Syst. Appl.*, **39** (2012), 10016–10021. https://doi.org/10.1016/j.eswa.2012.01.211
18. M. Rowshannahad, S. Dauzère-Pérès, B. Cassini, Capacitated Qualification Management in Semiconductor Manufacturing, *Omega*, **54** (2015), 50–59. https://doi.org/10.1016/j.omega.2015.01.012
19. A. Perraudat, S. Dauzère-Pérès, P. Vialletelle, Robust Tactical Qualification Decisions in Flexible Manufacturing Systems, *Omega*, **106** (2022), 102537. https://doi.org/10.1016/j.omega.2021.102537

20. A. Perraudat, S. Dauzère-Pérès, P. Vialletelle, Optimizing Multiple Qualifications of Products on Non-Identical Parallel Machines, *Comput. Oper. Res.*, **144** (2022), 105813. https://doi.org/10.1016/j.cor.2022.105813
21. J. Feng, H. Li, W. T. Huh, M. C. Sampaio, N. Keng, Erratum for "Planning Production and Equipment Qualification under High Process Flexibility", *Prod. Oper. Manag.*, **31** (2022), 1890–1892. https://doi.org/10.1111/poms.13682
22. K. Tamssaouet, S. Dauzère-Pérès, On the Impact of Operation Flexibility in the Flexible Job-Shop Scheduling Problem, *Omega*, **144** (2026), 103591. https://doi.org/10.1016/j.omega.2026.103591
23. M. Nattaf, S. Dauzère-Pérès, C. Yugma, C. Wu, Parallel Machine Scheduling with Time Constraints on Machine Qualifications, *Comput. Oper. Res.*, **107** (2019), 61–76. https://doi.org/10.1016/j.cor.2019.03.004
24. X. Chang, M. Dong, Product–Machine Qualification Using a Process Flexibility Strategy Considering Capacity Loss, *IISE Trans.*, **56** (2024), 98–113. https://doi.org/10.1080/24725854.2023.2202226
25. J. Chen, X. Ye, W. Ma, D. Xu, Logic-Based Benders Decomposition for Order Acceptance and Scheduling on Heterogeneous Factories with Carbon Caps, *Comput. Oper. Res.*, **168** (2024), 106706. https://doi.org/10.1016/j.cor.2024.106706
26. J. Chen, X. Ye, N. Xie, Y. Sang, M. Sterna, Benders Decompositions for Order Acceptance and Scheduling in Additive Manufacturing, *Eur. J. Oper. Res.*, **330** (2026), 745–759. https://doi.org/10.1016/j.ejor.2025.10.005
27. C. Zheng, J. Yu, G. Wan, Online order acceptance and scheduling in a single machine environment, *Comput. Oper. Res.*, **179** (2025), 107028. https://doi.org/10.1016/j.cor.2025.107028
28. B. Ervural, A. Özaydın, Integrated order acceptance and inventory policy optimization in a multi-period, multi-product hybrid production system, *Oper. Res. Perspect.*, **13** (2024), 100318. https://doi.org/10.1016/j.orp.2024.100318
29. G. Xu, N. Wu, Y. Qiao, S. Zhang, Exact approach to joint order acceptance and non-permutation flow shop scheduling via logic-based Benders decomposition, *Int. J. Prod. Res.*, (2026), 1–27. https://doi.org/10.1080/00207543.2026.2739603
30. T. Aouam, N. Brahimi, Integrated Production Planning and Order Acceptance under Uncertainty: A Robust Optimization Approach, *Eur. J. Oper. Res.*, **228** (2013), 504–515. https://doi.org/10.1016/j.ejor.2013.02.010
31. Y. Boulaksil, M. Grunow, J. C. Fransoo, Capacity Flexibility Allocation in an Outsourced Supply Chain with Reservation, *Int. J. Prod. Econ.*, **129** (2011), 111–118. https://doi.org/10.1016/j.ijpe.2010.09.010
32. E. K. Bish, A. Muriel, S. Biller, Managing Flexible Capacity in a Make-to-Order Environment, *Manag. Sci.*, **51** (2005), 167–180. https://doi.org/10.1287/mnsc.1040.0322
33. A. A. Syntetos, M. Z. Babai, N. Altay, On the Demand Distributions of Spare Parts, *Int. J. Prod. Res.*, **50** (2012), 2101–2117. https://doi.org/10.1080/00207543.2011.562561
34. S. Moon, C. Hicks, A. Simpson, The Development of a Hierarchical Forecasting Method for Predicting Spare Parts Demand in the South Korean Navy—A Case Study, *Int. J. Prod. Econ.*, **140** (2012), 794–802. https://doi.org/10.1016/j.ijpe.2012.02.012

35. S. Moon, A. Simpson, C. Hicks, The Development of a Classification Model for Predicting the Performance of Forecasting Methods for Naval Spare Parts Demand, *Int. J. Prod. Econ.*, **143** (2013), 449–454. https://doi.org/10.1016/j.ijpe.2012.02.016
36. L. Berterottière, S. Dauzère-Pérès, C. Yugma, Flexible Job-Shop Scheduling with Transportation Resources, *Eur. J. Oper. Res.*, **312** (2024), 890–909. https://doi.org/10.1016/j.ejor.2023.07.036
37. Y. Dong, S. Jiang, L. Zhou, Z. Jiang, Y. Yu, A Joint Scheduling Approach for Production and Material Handling under Customized Manufacturing Paradigm, *Eur. J. Oper. Res.*, **332** (2026), 730–747. https://doi.org/10.1016/j.ejor.2026.01.022
38. J. N. Hooker, Planning and Scheduling by Logic-Based Benders Decomposition, *Oper. Res.*, **55** (2007), 588–602. https://doi.org/10.1287/opre.1060.0371
39. Özgün Elçi, J. N. Hooker, Stochastic Planning and Scheduling with Logic-Based Benders Decomposition, *INFORMS J. Comput.*, **34** (2022), 2428–2442. https://doi.org/10.1287/ijoc.2022.1184
40. G. Codato, M. Fischetti, Combinatorial Benders' Cuts for Mixed-Integer Linear Programming, *Oper. Res.*, **54** (2006), 756–766. https://doi.org/10.1287/opre.1060.0286
41. A. H. C. Eaves, B. G. Kingsman, Forecasting for the Ordering and Stock-Holding of Spare Parts, *J. Oper. Res. Soc.*, **55** (2004), 431–437. https://doi.org/10.1057/palgrave.jors.2601697
42. Z. Gu, G. L. Nemhauser, M. W. P. Savelsbergh, Lifted Cover Inequalities for 0-1 Integer Programs: Computation, *INFORMS J. Comput.*, **10** (1998), 427–437. https://doi.org/10.1287/ijoc.10.4.427
43. Z. Gu, G. L. Nemhauser, M. W. P. Savelsbergh, Lifted Cover Inequalities for 0-1 Integer Programs: Complexity, *INFORMS J. Comput.*, **11** (1999), 117–123. https://doi.org/10.1287/ijoc.11.1.117
44. M. Queyranne, Structure of a Simple Scheduling Polyhedron, *Math. Program.*, **58** (1993), 263–285. https://doi.org/10.1007/BF01581271
45. M. Queyranne, Y. Wang, Single-Machine Scheduling Polyhedra with Precedence Constraints, *Math. Oper. Res.*, **16** (1991), 1–20. https://doi.org/10.1287/moor.16.1.1
46. P. Baptiste, C. Le Pape, W. P. M. Nuijten, Satisfiability Tests and Time-Bound Adjustments for Cumulative Scheduling Problems, *Ann. Oper. Res.*, **92** (1999), 305–333. https://doi.org/10.1023/A:1018995000688
47. J. N. Hooker, G. Ottosson, Logic-Based Benders Decomposition, *Math. Program.*, **96** (2003), 33–60. https://doi.org/10.1007/s10107-003-0375-9
48. V. Deshpande, A. V. Iyer, R. Cho, Efficient Supply Chain Management at the U.S. Coast Guard Using Part-Age Dependent Supply Replenishment Policies, *Oper. Res.*, **54** (2006), 1028–1040. https://doi.org/10.1287/opre.1060.0327
49. J. C. García-Benito, M. Martín-Peña, A Redistribution Model with Minimum Backorders of Spare Parts: A Proposal for the Defence Sector, *Eur. J. Oper. Res.*, **291** (2021), 178–193. https://doi.org/10.1016/j.ejor.2020.09.014




AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0)