Monotone Co-Design with Adaptive Optimistic Sampling

Meshal Alharbi, Munther A. Dahleh, Gioele Zardini

Abstract— **When designing systems with multiple conflicting objectives, efficiently discovering the full set of non-dominated solutions remains a fundamental challenge. In this paper, we introduce an online learning framework for multi-objective decision-making within the monotone co-design setting. An agent sequentially queries an expensive design problem to identify the set of non-dominated trade-offs while minimizing evaluations. The key mechanism is a generic family of optimistic evaluators, history-dependent bounds that enable safe early elimination of unpromising implementations and allow heterogeneous structural assumptions. We prove that the resulting rejection-sampling algorithm is sound and that the admissible region contracts monotonically as data accumulates, and instantiate the evaluators under monotonicity, Lipschitz continuity, and linear parametric structure. Experiments on intermodal mobility systems and synthetic benchmarks demonstrate substantial gains over uniform sampling, Bayesian optimization, and evolutionary baselines.**

I. Introduction

Many engineering domains require simultaneous optimization of conflicting objectives—such as cost versus reliability, or efficiency versus safety—where no single solution dominates all others. Instead, decision-makers must reason over the full set of Pareto-optimal trade-offs. This challenge is especially acute in systems composed of tightly coupled subsystems, as arise in robotics, autonomous mobility, and infrastructure planning, where component-level or scalarized optimization can mask important trade-offs and produce fragile designs [1], [2].

Solving multi-objective problems at scale is difficult for several reasons: coupling among subsystems creates large, interdependent search spaces; recovering the full efficient set is computationally demanding even when models are available [3], [4]. Monotone co-design theory offers a principled framework for this purpose [2], [5], [6]. By describing each subsystem through interfaces of functionalities and resources, modeled as partially ordered sets, it preserves the native order structure of the quantities being traded and enables system-level reasoning through structured operations on smaller blocks. However, existing methods for monotone co-design assume that every subsystem is available in explicit, tractable form. In practice, the most critical subsystem is often accessible only through expensive simulation or hardware-in-the-loop experiments. This motivates an *adaptive sampling* approach that directs evaluations toward the most informative candidates, using available structure to prune regions that provably cannot contribute to optimal trade-offs.

The authors are with the Laboratory for Information & Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 {meshal,dahleh,gzardini}@mit.edu

Alharbi and Zardini were supported by DARPA YFA (D25AC00373).

This paper studies online Pareto-set identification for co-design problems with expensive black-box subsystems. From a systems-and-control perspective, the problem is to allocate a limited budget of system-level simulations or experiments when one subsystem inside an interconnected design problem is only accessible through expensive oracle calls. The novelty is a co-design-native elimination principle: history-dependent optimistic evaluators use partial-order structure to safely discard implementations before evaluation, rather than relying on scalarization, a fixed stochastic reward model, or a globally metric formulation of the problem.

Our contributions are as follows. First, we define an online learning problem over partially ordered functionality and resource spaces, where the goal is to recover the antichain of non-dominated resources compatible with a desired functionality target. Second, we introduce generic optimistic evaluators, establish safe elimination and monotone shrinking properties, and instantiate this template under common structural assumptions. Finally, we develop an elimination-based rejection sampler with rigorous correctness guarantees and evaluate it on a realistic intermodal mobility co-design problem and synthetic benchmarks with Lipschitz structure, demonstrating substantial gains over standard baselines.

A. Related Literature

Approximating Pareto fronts in expensive black-box settings is a long-standing challenge addressed by several complementary paradigms. Multi-objective Evolutionary Algorithms (EAs) maintain populations updated via selection and mutation, guided either by non-dominated sorting or by decomposition into scalar subproblems [7]–[9]. Bayesian optimization methods instead build probabilistic surrogates and select evaluations via acquisition criteria—scalarization [10], hypervolume improvement [11], or information-theoretic gains [12]. Multi-objective bandits pursue sample-complexity guarantees for Pareto set identification using optimistic confidence sets [13], [14], with recent extensions to general preference cones and structured linear models [15], [16], though sequential-testing approaches can incur high computational costs [17].

Our work differs from these lines of research in that it is formulated on general *partially ordered* functionality and resource spaces, without requiring scalarization or metric structure in the core problem formulation. The overall system is treated order-theoretically, and additional structure, such as monotonicity or metric smoothness, is optional and enters only through specific optimistic evaluators. This separation allows heterogeneous structural assumptions to be integrated modularly within a single unified framework.

II. Preliminaries

A. Background on Order

To start, we summarize the main order-theoretic notions used throughout this paper.

Definition II.1 (Poset)**.** A *partially ordered set (poset)* is a tuple $\mathcal{P} = \langle P, \preceq_{\mathcal{P}} \rangle$, where P is a set and $\preceq_{\mathcal{P}}$ is a partial order (a binary relation on P that is reflexive, antisymmetric, and transitive).

For a poset $\mathcal{P} = \langle P, \preceq_{\mathcal{P}} \rangle$, we write $x \prec_{\mathcal{P}} y$ when $x \preceq_{\mathcal{P}} y$ and $x \neq y$. A *total order* is a poset in which every pair of elements is comparable.

Definition II.2 (Opposite poset)**.** The *opposite* of a poset $\mathcal{P} = \langle P, \preceq_{\mathcal{P}} \rangle$ is the poset $\mathcal{P}^{\mathrm{op}} := \langle P, \preceq_{\mathcal{P}}^{\mathrm{op}} \rangle$ with the same elements and reversed ordering: $x \preceq_{\mathcal{P}}^{\mathrm{op}} y \Leftrightarrow y \preceq_{\mathcal{P}} x$.

Definition II.3 (Product poset)**.** Given two posets $\mathcal{P} = \langle P, \preceq_{\mathcal{P}} \rangle$ and $\mathcal{Q} = \langle Q, \preceq_{\mathcal{Q}} \rangle$, their *product* $\mathcal{P} \times \mathcal{Q} = \langle P \times Q, \preceq_{\mathcal{P}\times\mathcal{Q}} \rangle$ is a poset with: $\langle x_P, x_Q \rangle \preceq_{\mathcal{P}\times\mathcal{Q}} \langle y_P, y_Q \rangle \Leftrightarrow (x_P \preceq_{\mathcal{P}} y_P) \wedge (x_Q \preceq_{\mathcal{Q}} y_Q)$.

Definition II.4 (Antichain in a poset)**.** An *antichain* is a subset S of a poset $\mathcal{P}$ where no two distinct elements in S are comparable: $(x, y \in S) \wedge (x \preceq_{\mathcal{P}} y) \Rightarrow x = y$. We denote the set of antichains of a poset $\mathcal{P}$ by $\mathsf{Anti}\,\mathcal{P}$.

For an arbitrary poset $\mathcal{P}$, the canonical comparison between two antichains $A, B \in \mathsf{Anti}\,\mathcal{P}$ is via dominated regions: we say that A *dominates* B if $\uparrow A \supseteq \uparrow B$.

Definition II.5 (Upper closure)**.** Given a poset $\mathcal{P}$, the *upper closure* of a subset $S \subseteq \mathcal{P}$ is a subset $\uparrow S \subseteq \mathcal{P}$ that contains all elements y of $\mathcal{P}$ that are greater or equal to some element x in S: $\uparrow S = \{y \in P \mid \exists x \in S : x \preceq_{\mathcal{P}} y\}$.

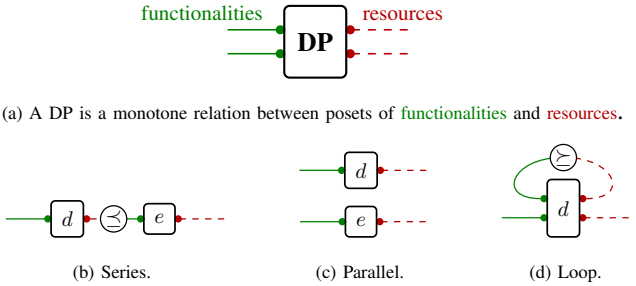
Definition II.6 (Monotone map)**.** A map $f : \mathcal{P} \to \mathcal{Q}$ between posets $\langle P, \preceq_{\mathcal{P}} \rangle$ and $\langle Q, \preceq_{\mathcal{Q}} \rangle$ is *monotone* if $x \preceq_{\mathcal{P}} y$ implies $f(x) \preceq_{\mathcal{Q}} f(y)$. Monotonicity is preserved by composition and products.

For a subset $S \subseteq P$ of a poset $\mathcal{P}$, we write $\min_{\preceq_{\mathcal{P}}} S := \{x \in S \mid \nexists y \in S : y \prec_{\mathcal{P}} x\}$ for the set of minimal elements of S, and $\max_{\preceq_{\mathcal{P}}} S := \{x \in S \mid \nexists y \in S : x \prec_{\mathcal{P}} y\}$ for the set of maximal elements. Whenever nonempty, both $\min_{\preceq_{\mathcal{P}}} S$ and $\max_{\preceq_{\mathcal{P}}} S$ are antichains. When $\mathcal{P}$ is a lattice, we write $\bigvee S$ and $\bigwedge S$ for the join (supremum) and meet (infimum) of S, respectively, whenever they exist.

B. Background on Co-Design

Co-design provides a compositional language for reasoning about systems with heterogeneous interfaces and coupled design decisions. Its basic objects are *functionalities* and *resources*, modeled as posets. Informally, more functionality is preferred, whereas fewer resources are preferred.

Definition II.7 (Design Problem (DP))**.** Given posets $\mathcal{F}$ and $\mathcal{R}$ of functionalities and resources, a *Design Problem (DP)* is an upper set of $\mathcal{F}^{\mathrm{op}} \times \mathcal{R}$. We denote the set of such DPs by $\mathsf{DP}\{\mathcal{F}, \mathcal{R}\}$. Given a DP dp, a pair $\langle f, r \rangle$ of functionality f and resource r is *feasible* if $\langle f, r \rangle \in \mathrm{dp}$. We order $\mathsf{DP}\{\mathcal{F}, \mathcal{R}\}$ by inclusion: $\mathrm{dp}_a \preceq \mathrm{dp}_b \Leftrightarrow \mathrm{dp}_a \subseteq \mathrm{dp}_b$.



(a) A DP is a monotone relation between posets of functionalities and resources.

(b) Series. (c) Parallel. (d) Loop.

Fig. 1. DPs can be composed in different ways.

Definition II.8 (Design Problem with Implementation (DPI))**.** Given posets $\mathcal{F}$ and $\mathcal{R}$ of functionalities and resources, a *Design Problem with Implementation (DPI)* is a tuple $\langle \mathcal{I}, \mathsf{prov}, \mathsf{req} \rangle$, with a set of implementations $\mathcal{I}$ and maps $\mathsf{prov} : \mathcal{I} \to \mathcal{F}$ and $\mathsf{req} : \mathcal{I} \to \mathcal{R}$. For each design choice $i \in \mathcal{I}$, $\mathsf{prov}(i)$ represents the functionality provided by i, while $\mathsf{req}(i)$ represents the resource it requires. We denote the set of such DPIs by $\mathsf{DPI}\{\mathcal{I}, \mathcal{F}, \mathcal{R}\}$.

For each DPI, there is a corresponding DP given by the free choice among all implementations: $\uparrow\{\langle \mathsf{prov}(i), \mathsf{req}(i) \rangle \mid i \in \mathcal{I}\} \in \mathsf{DP}\{\mathcal{F}, \mathcal{R}\}$.

Definition II.9 (Queries for DPI)**.** Suppose we have a DPI $\langle \mathcal{I}, \mathsf{prov}, \mathsf{req} \rangle$, whose functionality and resource poset are $\mathcal{F}$ and $\mathcal{R}$, respectively. We define the *Fix functionality minimize resources* query: For a fixed functionality $f \in \mathcal{F}$, return all implementations i that satisfies f and minimize the resources (i.e., $\mathsf{req}(i) \in \min_{\preceq_{\mathcal{R}}} \{\mathsf{req}(j) \mid f \preceq_{\mathcal{F}} \mathsf{prov}(j), j \in \mathcal{I}\}$. We denote the set of minimal resources for this query by $\mathsf{FixFunMinRes}(f) \in \mathsf{Anti}\,\mathcal{R}$.

A DP is a generalization of a black-box multi-objective problem [1], [6], where one only models the space of design choices $\mathcal{I}$ and a mapping req from design choices to objectives. In DPs, we also model what satisfying the requirements provides through the functionality space $\mathcal{F}$ and the map prov, which allows us to compose a DP with other DPs to construct larger problems in a modular way (Figure 1).

III. Problem Formulation

This section introduces an online framework for sequential decision-making for DP. We consider an agent that interacts with a DPI $\mathrm{dp} \in \mathsf{DPI}\{\mathcal{I}, \mathcal{F}, \mathcal{R}\}$ in an online, adaptive manner, sequentially querying dp and adjusting decisions based on observed outcomes. Since the problem involves posets $\mathcal{F}$ and $\mathcal{R}$, the goal is to identify a *set* of non-dominated solutions. We impose the following assumptions.

Assumption 1 (Top and bottom elements)**.** The posets $\mathcal{F}$ and $\mathcal{R}$ each have top and bottom elements $\top_{\mathcal{F}}, \bot_{\mathcal{F}}$ and $\top_{\mathcal{R}}, \bot_{\mathcal{R}}$, respectively.

Assumption 2 (Feasibility and upward generation by minima)**.** For any $f \in \mathcal{F}$, there exists a nonempty subset $\mathcal{I}_f \subseteq \mathcal{I}$

such that $f \preceq_{\mathcal{F}} \mathsf{prov}(i)$ for all $i \in \mathcal{I}_f$. Moreover, the set of minimal elements $\mathsf{FixFunMinRes}(f)$ exists, and every target-feasible resource is above one of these minimal elements.

Assumption 3 (Sampling measure)**.** The implementation space $\mathcal{I}$ is a Polish space equipped with a full-support Borel probability measure μ, i.e., $\mu(U) > 0$ for every non-empty open set $U \subseteq \mathcal{I}$.

Assumption 1 is mild: one can always augment $\mathcal{F}$ and $\mathcal{R}$ with artificial extremal elements. We will use the extremal elements as a convenient way to define default behavior for algorithms. Assumption 2 ensures the object of interest is well-posed, so the agent need not reason about infeasibility certificates. Assumption 3 guarantees that i.i.d. draws from μ eventually explore every region of $\mathcal{I}$, subsuming both finite spaces with strictly positive distributions and bounded subsets of $\mathbb{R}^d$ with normalized Lebesgue measure, without imposing further structure on $\mathcal{I}$. Now, the online learning problem is as follows:

Problem III.1 (Online design)**.** Given a target functionality $f \in \mathcal{F}$ and an evaluation budget N, the agent interacts with a DPI $\mathrm{dp} \in \mathsf{DPI}\{\mathcal{I}, \mathcal{F}, \mathcal{R}\}$ by selecting implementations $i \in \mathcal{I}$ and observing $\mathsf{prov}(i)$ and $\mathsf{req}(i)$. The goal is to return a set $I_N \subseteq \mathcal{I}$, $|I_N| \leq N$, whose induced resource antichain recovers the true solution $\mathsf{FixFunMinRes}(f)$.

If two implementations share the same nondominated resource value, returning either suffices. We now formalize induced antichains. At time $t \in [1, N]$, the agent's history is $H_t := \{\langle i, \mathsf{prov}(i), \mathsf{req}(i)\rangle \mid i \in I_t\}$, where $I_t \subseteq \mathcal{I}$ collects all implementations queried so far. The set of all finite histories is $\mathcal{H} := \mathsf{Finite}\big(\{\langle i, \mathsf{prov}(i), \mathsf{req}(i)\rangle \mid i \in \mathcal{I}\}\big)$, which is a subset of all finite triples of $\mathcal{I} \times \mathcal{F} \times \mathcal{R}$. We define projections $\Pi_{\mathcal{R}} \colon \mathcal{H} \to \mathsf{Pow}(\mathcal{R})$, $\Pi_{\mathcal{I}} \colon \mathcal{H} \to \mathsf{Pow}(\mathcal{I})$, and $\Pi_{\mathcal{F}} \colon \mathcal{H} \to \mathsf{Pow}(\mathcal{F})$ that extract the resource, implementation, and functionality components of a history, respectively.

Definition III.1 (History-induced antichain)**.** Given a history $H \in \mathcal{H}$, the target-feasible induced resource antichain is

$$\mathsf{Anti}_f(H) := \min_{\preceq_{\mathcal{R}}} \Pi_{\mathcal{R}}(\{\langle i, f', r\rangle \in H \mid f \preceq_{\mathcal{F}} f'\}). \quad (1)$$

After N queries, the agent returns I_N inducing the approximate antichain $\hat{A}_f := \mathsf{Anti}_f(H_N)$. With $A_f := \mathsf{FixFunMinRes}(f)$, exact recovery occurs when $\hat{A}_f = A_f$. In the general partially ordered setting, requiring every point on the true antichain to be covered within a specified tolerance would need additional structure on $\mathcal{R}$.

IV. Methodology and Algorithm

In this section, we develop an elimination-based algorithm for Problem III.1. The core idea is to equip the agent with history-dependent bounds on the unknown maps req and prov that can certify certain unqueried implementations as irrelevant to the current target-feasible nondominated set, allowing them to be discarded without evaluation. We first introduce optimistic evaluators, then present a rejection-sampling algorithm, and then describe structural settings in which such evaluators can be constructed explicitly.

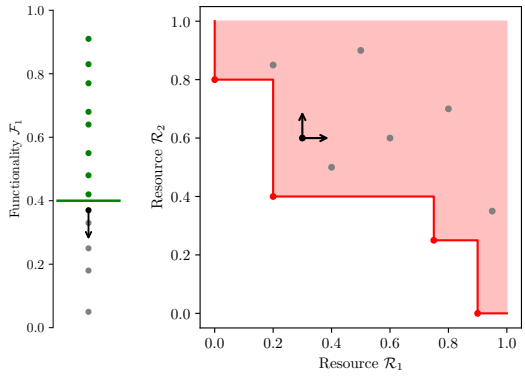


Fig. 2. Example for an implementation i that can be eliminated using either of the optimistic evaluators. Left: The functionality space ($\mathcal{F} = \mathbb{R}$) with $f = 0.4$. The green points satisfy the functionality, while the gray points do not. The black point with an arrow shows the value of $\mathsf{prov}^{\mathrm{opt}}(i, H)$. Right: The resources space ($\mathcal{R} = \mathbb{R}^2$) with the upper closure of the current antichain highlighted. The red points are non-dominated, while the gray points are dominated. The black point with arrows shows the value of $\mathsf{req}^{\mathrm{opt}}(i, H)$.

A. Optimistic Evaluators

Definition IV.1 (Optimistic evaluators)**.** Maps $\mathsf{prov}^{\mathrm{opt}} \colon \mathcal{I} \times \mathcal{H} \to \mathcal{F}$ and $\mathsf{req}^{\mathrm{opt}} \colon \mathcal{I} \times \mathcal{H} \to \mathcal{R}$ are called *optimistic evaluators* if, for every $i \in \mathcal{I}$ and $H \in \mathcal{H}$,

$$\mathsf{req}^{\mathrm{opt}}(i, H) \preceq_{\mathcal{R}} \mathsf{req}(i), \quad (2)$$

$$\mathsf{prov}(i) \preceq_{\mathcal{F}} \mathsf{prov}^{\mathrm{opt}}(i, H). \quad (3)$$

That is, $\mathsf{req}^{\mathrm{opt}}$ lower-bounds the required resources and $\mathsf{prov}^{\mathrm{opt}}$ upper-bounds the provided functionality. The term *optimistic* reflects the fact that resources are minimized and functionalities maximized, so these bounds represent the most favorable assessment of i consistent with H. By Assumption 1, trivial optimistic evaluators always exist (e.g., $\mathsf{req}^{\mathrm{opt}} \equiv \bot_{\mathcal{R}}$), but such bounds are uninformative. The history dependence allows bounds to tighten as data accumulates: one expects $\mathsf{req}^{\mathrm{opt}}(i, H)$ to increase and $\mathsf{prov}^{\mathrm{opt}}(i, H)$ to decrease as H grows, yielding progressively sharper elimination. Concrete instances are given in Section IV-C.

We now show how these evaluators enable elimination. Fix a target $f \in \mathcal{F}$, a history $H \in \mathcal{H}$, and an unqueried implementation $i \in \mathcal{I} \setminus \Pi_{\mathcal{I}}(H)$. Implementation i can be safely discarded if either:

$$\mathsf{req}^{\mathrm{opt}}(i, H) \in \uparrow \mathsf{Anti}_f(H), \quad (4)$$

$$\mathsf{prov}^{\mathrm{opt}}(i, H) \prec_{\mathcal{F}} f. \quad (5)$$

Condition (4) means that even the most favorable resource estimate is already dominated by the current antichain, while (5) means even the most favorable functionality estimate fails to meet the target. By (2) and (3), these imply $\mathsf{req}(i) \in \uparrow \mathsf{Anti}_f(H)$ or $f \not\preceq_{\mathcal{F}} \mathsf{prov}(i)$, respectively, so i cannot contribute to the solution and may be discarded without evaluation. This elimination principle underlies the algorithm below; a graphical illustration appears in Figure 2, where the Euclidean structure is used solely for visualization purposes and is not required by the framework.

Algorithm 1 Rejection Sampler with Optimistic Evaluators

input: Target functionality f and evaluation budget N
initialize: History $H = \{\}$
for $t = 1, \dots, N$ **do**
  Set Accept $= \bot$
  while $\neg$Accept **do**
    Sample $i \sim \mu(\mathcal{I} \setminus \Pi_{\mathcal{I}}(H))$
    Evaluate $r^{\mathrm{opt}} = \mathsf{req}^{\mathrm{opt}}(i, H)$ and $f^{\mathrm{opt}} = \mathsf{prov}^{\mathrm{opt}}(i, H)$
    Set Accept $= \top$ if $(r^{\mathrm{opt}} \notin \uparrow \mathsf{Anti}_f(H)) \wedge (f \preceq_{\mathcal{F}} f^{\mathrm{opt}})$
  query expensive block:
  Query $r = \mathsf{req}(i)$ and $f' = \mathsf{prov}(i)$ from dp
  Append $\langle i, f', r \rangle$ to the history H
return: Antichain $\mathsf{Anti}_f(H)$ and its corresponding implementations

B. Rejection Sampling with Optimistic Evaluations

We now describe an elimination-based procedure that uses optimistic evaluators to focus sampling on implementations that may still improve the current solution estimate.

Definition IV.2 (Admissible set)**.** Fix a target functionality $f \in \mathcal{F}$. For a history $H \in \mathcal{H}$, the *admissible set* of implementations $\mathcal{A}(H) \subseteq \mathcal{I}$ is $\mathcal{A}(H) := \{i \in \mathcal{I} \setminus \Pi_{\mathcal{I}}(H) \mid \mathsf{req}^{\mathrm{opt}}(i, H) \notin \uparrow \mathsf{Anti}_f(H), f \preceq_{\mathcal{F}} \mathsf{prov}^{\mathrm{opt}}(i, H)\}$.

Thus, $\mathcal{A}(H)$ consists precisely of those unqueried implementations that cannot yet be ruled out as irrelevant to the target query. Equivalently, it is the set of implementations that remain plausible candidates for improving the current target-feasible antichain.

Algorithm 1 may be viewed as a rejection sampler over the current admissible set. At each round, the algorithm draws implementations from the base measure μ until it finds one that is not eliminated by the optimistic tests, and only then queries the true values of req and prov. In this way, the algorithm concentrates its budget on the subset of implementations that remain potentially relevant under the information encoded in the history. By design, Algorithm 1 accommodates *heterogeneous spaces*; for instance, the resource space $\mathcal{R}$ may be a cone-ordered vector space, whereas the functionality space $\mathcal{F}$ may be equipped only with a discrete partial order.

To ensure the rejection-sampling loop is well defined up to budget N, we assume that the admissible set is always nonempty and has positive μ-measure along histories of interest.

Assumption 4 (Well-defined rejection sampling)**.** For every history $H \in \mathcal{H}$, the admissible set $\mathcal{A}(H)$ is μ-measurable with $\mu(\mathcal{A}(H)) > 0$.

Now, we list formal guarantees on our algorithm. The next lemma formalizes the *soundness* of the elimination rule.

Lemma IV.1 (Safe elimination by optimistic evaluators)**.** Let $H \in \mathcal{H}$ and $i \in \mathcal{I}$. Assume either $\mathsf{req}^{\mathrm{opt}}(i, H) \in \uparrow \mathsf{Anti}_f(H)$ or $f \not\preceq_{\mathcal{F}} \mathsf{prov}^{\mathrm{opt}}(i, H)$. Then:

$$\neg\Big(f \preceq_{\mathcal{F}} \mathsf{prov}(i) \ \wedge \ \mathsf{req}(i) \notin \uparrow \mathsf{Anti}_f(H)\Big). \tag{6}$$

Equivalently, i cannot be both target-feasible and an improvement over the current feasible antichain.

Proof. We prove each case separately.

Case 1: Suppose that $\mathsf{req}^{\mathrm{opt}}(i, H) \in \uparrow \mathsf{Anti}_f(H)$. By definition of upper closure, there exists some $r_H \in \mathsf{Anti}_f(H)$ such that $r_H \preceq_{\mathcal{R}} \mathsf{req}^{\mathrm{opt}}(i, H)$. By optimism (2), $\mathsf{req}^{\mathrm{opt}}(i, H) \preceq_{\mathcal{R}} \mathsf{req}(i)$. By transitivity, $r_H \preceq_{\mathcal{R}} \mathsf{req}(i)$, hence $\mathsf{req}(i) \in \uparrow \mathsf{Anti}_f(H)$.

Case 2: Suppose that $f \not\preceq_{\mathcal{F}} \mathsf{prov}^{\mathrm{opt}}(i, H)$. By optimism (3), $\mathsf{prov}(i) \preceq_{\mathcal{F}} \mathsf{prov}^{\mathrm{opt}}(i, H)$. If $f \preceq_{\mathcal{F}} \mathsf{prov}(i)$, then by transitivity $f \preceq_{\mathcal{F}} \mathsf{prov}^{\mathrm{opt}}(i, H)$, a contradiction. Hence $f \not\preceq_{\mathcal{F}} \mathsf{prov}(i)$. □

The next result shows that, under a natural monotonicity condition, the admissible set *can only shrink* as more information is collected.

Lemma IV.2 (Monotone shrinking of the admissible set)**.** Let $i \in \mathcal{I}$ and $H, H' \in \mathcal{H}$ such that $H \subseteq H'$. Assume that:

$$\begin{aligned} \mathsf{req}^{\mathrm{opt}}(i, H) &\preceq_{\mathcal{R}} \mathsf{req}^{\mathrm{opt}}(i, H'), \\ \mathsf{prov}^{\mathrm{opt}}(i, H') &\preceq_{\mathcal{F}} \mathsf{prov}^{\mathrm{opt}}(i, H). \end{aligned} \tag{7}$$

Then $\mathcal{A}(H') \subseteq \mathcal{A}(H)$.

Proof. Since $H \subseteq H'$, we have $\uparrow \mathsf{Anti}_f(H) \subseteq \uparrow \mathsf{Anti}_f(H')$. Let $i \in \mathcal{A}(H')$. Then $\mathsf{req}^{\mathrm{opt}}(i, H') \notin \uparrow \mathsf{Anti}_f(H')$, which implies that $\mathsf{req}^{\mathrm{opt}}(i, H') \notin \uparrow \mathsf{Anti}_f(H)$. Now, by hypothesis (7), $\mathsf{req}^{\mathrm{opt}}(i, H) \preceq_{\mathcal{R}} \mathsf{req}^{\mathrm{opt}}(i, H')$. If $\mathsf{req}^{\mathrm{opt}}(i, H) \in \uparrow \mathsf{Anti}_f(H)$, then since the upper closure is an upper set and $\mathsf{req}^{\mathrm{opt}}(i, H) \preceq_{\mathcal{R}} \mathsf{req}^{\mathrm{opt}}(i, H')$, we would have $\mathsf{req}^{\mathrm{opt}}(i, H') \in \uparrow \mathsf{Anti}_f(H)$, contradicting what we just established. Hence $\mathsf{req}^{\mathrm{opt}}(i, H) \notin \uparrow \mathsf{Anti}_f(H)$.

Similarly, $f \preceq_{\mathcal{F}} \mathsf{prov}^{\mathrm{opt}}(i, H')$ and $\mathsf{prov}^{\mathrm{opt}}(i, H') \preceq_{\mathcal{F}} \mathsf{prov}^{\mathrm{opt}}(i, H)$ yield $f \preceq_{\mathcal{F}} \mathsf{prov}^{\mathrm{opt}}(i, H)$ by transitivity. Therefore $i \in \mathcal{A}(H)$. □

Now, we use these properties to establish antichain recovery results for Algorithm 1.

Theorem IV.1 (Trichotomy for optimal witnesses)**.** Consider Algorithm 1 without a sample budget under Assumptions 1 to 4, with valid optimistic evaluators (Definition IV.1). Let H_t denote the history after round t. Then for every $r^\star \in A_f$ and every witness $i^\star$ satisfying $f \preceq_{\mathcal{F}} \mathsf{prov}(i^\star)$ and $\mathsf{req}(i^\star) = r^\star$, almost surely exactly one of the following occurs:

(a) *(Queried.)* $i^\star$ is queried at some round t, in which case $r^\star \in \uparrow \mathsf{Anti}_f(H_t)$.
(b) *(Eliminated.)* $i^\star$ is never queried but is eliminated from $\mathcal{A}(H_t)$ at some round t, in which case $r^\star \in \uparrow \mathsf{Anti}_f(H_t)$.
(c) *(Persistently admissible.)* $i^\star$ is never queried and remains admissible at every round, i.e., $i^\star \in \mathcal{A}(H_t)$ for all $t \geq 1$.

In Cases (a) and (b), $r^\star$ is covered by the approximate antichain.

Proof. Fix $r^\star \in A_f$ and a witness $i^\star$. The three cases are mutually exclusive and exhaustive.

Case (a): Suppose $i^\star \in \Pi_{\mathcal{I}}(H_t)$ at some round t. Since $i^\star$ is target-feasible ($f \preceq_{\mathcal{F}} \mathsf{prov}(i^\star)$), its resource $r^\star = \mathsf{req}(i^\star)$ enters $\mathsf{Anti}_f(H_t)$, giving $r^\star \in \uparrow \mathsf{Anti}_f(H_t)$.

Case (b): Suppose $i^\star \notin \Pi_{\mathcal{I}}(H_t)$ for all t (never queried), but $i^\star \notin \mathcal{A}(H_t)$ at some round t (eliminated). By Lemma IV.1, $i^\star$ cannot be both target-feasible and an improvement over the current antichain. Since $i^\star$ is target-feasible by assumption, it follows that $\mathsf{req}(i^\star) \in \uparrow \mathsf{Anti}_f(H_t)$, i.e., $r^\star \in \uparrow \mathsf{Anti}_f(H_t)$ through another witness.

Case (c): Suppose $i^\star \notin \Pi_{\mathcal{I}}(H_t)$ and $i^\star \in \mathcal{A}(H_t)$ for all t. This is precisely the persistent-admissibility alternative. □

Proposition IV.1 (Exact recovery from positive-measure witness sets)**.** Under the assumptions of Theorem IV.1, additionally assume that A_f is finite and that for every $r^\star \in A_f$, the set

$$W(r^\star) := \{i \in \mathcal{I} : f \preceq_{\mathcal{F}} \mathsf{prov}(i),\ \mathsf{req}(i) = r^\star\} \tag{8}$$

is Borel-measurable and satisfies $\mu(W(r^\star)) > 0$. Then there exists a random $N_0 < \infty$ such that

$$\uparrow \mathsf{Anti}_f(H_t) = \uparrow A_f, \qquad \forall t \geq N_0, \quad \text{almost surely.} \tag{9}$$

Proof. Fix $r^\star \in A_f$ and let $W^\star := W(r^\star)$. We show that $r^\star$ is covered in finite time almost surely.

If there exists $i^\star \in W^\star$ falling under Case (a) or (b) of Theorem IV.1, then there exists a finite round t such that $r^\star \in \uparrow \mathsf{Anti}_f(H_t)$. It remains to rule out the event that $r^\star$ is never covered.

Suppose $r^\star \notin \uparrow \mathsf{Anti}_f(H_t)$ for all t. Then no witness $i^\star \in W^\star$ can fall under Case (a) or (b) of Theorem IV.1, since either outcome would cover $r^\star$. Hence every $i^\star \in W^\star$ falls under Case (c), so $W^\star \subseteq \mathcal{A}(H_t)$ for all t.

At every round t, the conditional probability of sampling some witness in $W^\star$ is

$$\Pr(i_t \in W^\star \mid H_{t-1}) = \frac{\mu(W^\star)}{\mu(\mathcal{A}(H_{t-1}))} \geq \mu(W^\star) > 0, \tag{10}$$

where we used $W^\star \subseteq \mathcal{A}(H_{t-1}) \subseteq \mathcal{I}$ and $\mu(\mathcal{I}) = 1$ (Assumption 3). On the event that $r^\star$ is never covered, this bound holds at every round, so $\sum_{t \geq 1} \Pr(i_t \in W^\star \mid H_{t-1}) = \infty$. By the conditional Borel–Cantelli lemma (Lévy), $i_t \in W^\star$ for some finite time almost surely, contradicting the assumption for Case (c) that $r^\star$ is never covered.

Therefore, for each $r^\star \in A_f$, only Cases (a) and (b) occur, so $r^\star \in \uparrow \mathsf{Anti}_f(H_t)$ at some finite random time. Since the queried set grows monotonically with t, the sequence $\{\uparrow \mathsf{Anti}_f(H_t)\}_{t \geq 1}$ is nondecreasing, so there exists a finite random time $T(r^\star)$ such that $r^\star \in \uparrow \mathsf{Anti}_f(H_t)$, $\forall t \geq T(r^\star)$, almost surely. Since A_f is finite, the random variable $N_0 := \max_{r^\star \in A_f} T(r^\star)$ is almost surely finite, giving $\uparrow A_f \subseteq \uparrow \mathsf{Anti}_f(H_N)$, $\forall\, N \geq N_0$, almost surely. The reverse inclusion holds for all $N \geq 1$: every target-feasible queried implementation has a resource vector in $\uparrow A_f$, so $\uparrow \mathsf{Anti}_f(H_N) \subseteq \uparrow A_f$. Combining the two inclusions proves the claim. □

Beyond correctness (i.e., Case (b) of Theorem IV.1), the value of non-trivial optimistic evaluators shows in (10), where the probability of sampling an optimal witness $i^\star$ increases as the set $\mathcal{A}(H_{t-1})$ shrinks. While tightening is guaranteed by Lemma IV.2, the statement is not quantitative, and quantitative statements would require extra assumptions on the partial orders $\mathcal{F}$ and $\mathcal{R}$. Finally, Algorithm 1 is intentionally a baseline procedure. When more structure on the implementation space is specified, one may replace pure rejection sampling with more efficient search mechanisms.

C. Examples of Optimism

In this section, we discuss several structural settings in which optimistic evaluators can be constructed explicitly. For clarity, we focus on the resource evaluator $\mathsf{req}^{\mathrm{opt}}$; corresponding upper-bound constructions for $\mathsf{prov}^{\mathrm{opt}}$ are obtained by dual arguments.

1) Monotonicity: Assume that $\mathcal{I}$ is itself a poset, $\mathcal{R}$ is a join-semilattice, and that $\mathsf{req} \colon \mathcal{I} \to \mathcal{R}$ is monotone, i.e., $j \preceq_{\mathcal{I}} i \Rightarrow \mathsf{req}(j) \preceq_{\mathcal{R}} \mathsf{req}(i)$, $\forall\, i, j \in \mathcal{I}$. Given a history $H \in \mathcal{H}$ and implementation $i \in \mathcal{I}$, define:

$$S(i, H) := \{r \mid (\langle j, \cdot, r\rangle \in H \wedge (j \preceq_{\mathcal{I}} i)\}. \tag{11}$$

If $S(i, H) = \emptyset$, set $\mathsf{req}^{\mathrm{opt}}(i, H) = \bot_{\mathcal{R}}$. Otherwise, the join of $S(i, H)$ is a valid optimistic evaluator:

$$\mathsf{req}^{\mathrm{opt}}(i, H) = \bigvee S(i, H), \tag{12}$$

If $\mathcal{R}$ is a general poset, $\max_{\preceq_{\mathcal{R}}} S$ is an antichain, and one would need an additional tie-breaking rule to pick a single maximal element.

2) Lipschitz Continuity: Assume that $\mathcal{R} \subseteq \mathbb{R}^m$ is equipped with the componentwise order, that $\mathcal{I}$ is a metric space with metric d, and that req is L-Lipschitz with respect to d and the ℓ_p norm: $\|\mathsf{req}(i) - \mathsf{req}(j)\|_p \leq L d(i, j)$, $\forall\, i, j \in \mathcal{I}$. Given a history $H \in \mathcal{H}$ and implementation $i \in \mathcal{I}$, define:

$$S(i, H) = \{r - L d(i, j)\mathbf{1} \mid \langle j, \cdot, r\rangle \in H\} \tag{13}$$

If $S(i, H) = \emptyset$, set $\mathsf{req}^{\mathrm{opt}}(i, H) = \bot_{\mathcal{R}}$. Otherwise, define

$$\mathsf{req}^{\mathrm{opt}}(i, H) = \bigvee_{\leq_{\mathcal{R}}} S(i, H), \tag{14}$$

where $\leq_{\mathcal{R}}$ is the componentwise order. If desired, this lower bound may additionally be clipped below by $\bot_{\mathcal{R}}$. This construction is the same "optimism via smoothness" idea used in Lipschitz bandits [14], [18].

3) Linear-Parametric Models: As a third example, assume that $\mathcal{R} \subseteq \mathbb{R}^m$ is equipped with the componentwise order and that the resource map req belongs to a known linear-parametric class. Specifically, suppose there exist a known feature map $\Phi \colon \mathcal{I} \to \mathbb{R}^{m \times p}$ and an unknown parameter vector $\theta^\star \in \mathbb{R}^p$ such that:

$$\mathsf{req}(i) = \Phi(i)\, \theta^\star, \quad \forall\, i \in \mathcal{I}. \tag{15}$$

Further assume that the agent is given a prior set $\Theta_0 \subseteq \mathbb{R}^p$ such that $\theta^\star \in \Theta_0$. For a history $H \in \mathcal{H}$, define the

corresponding *confidence set* as the subset of Θ_0 consistent with all observations:

$$\Theta(H) = \{\theta \in \Theta_0 \mid \Phi(j)\,\theta = r \;\; \forall \langle j, \cdot, r\rangle \in H\}. \tag{16}$$

By construction, $\theta^\star \in \Theta(H)$ for every $H \in \mathcal{H}$, and $\Theta(H') \subseteq \Theta(H)$ whenever $H \subseteq H'$. Given a history $H \in \mathcal{H}$ and an implementation $i \in \mathcal{I}$, define:

$$S(i, H) = \{\Phi(i)\,\theta \mid \theta \in \Theta(H)\}. \tag{17}$$

Assuming that the coordinatewise infimum of $S(i, H)$ belongs to $\mathcal{R}$, define

$$\mathsf{req}^{\mathrm{opt}}(i, H) = \bigwedge_{\leq_{\mathcal{R}}} S(i, H), \tag{18}$$

where $\bigwedge_{\leq_{\mathcal{R}}}$ denotes the componentwise infimum (meet). Equivalently, each coordinate is obtained by solving a separate optimization:

$$[\mathsf{req}^{\mathrm{opt}}(i, H)]_k = \min_{\theta \in \Theta(H)} [\Phi(i)\,\theta]_k \qquad \forall\, k \in [m]. \tag{19}$$

When Θ_0 is a convex polytope, the confidence set $\Theta(H)$ is also a convex polytope (as the intersection of Θ_0 with affine constraints), and each coordinate in (19) reduces to a Linear Program (LP).

The three examples in this subsection rely on exact structure. If model misspecification is a concern, one may combine elimination with forced exploration, for example, by querying from the base measure μ with a small probability $\delta > 0$ and using the rejection sampler otherwise. This prevents a single erroneous optimistic certificate from permanently excluding an implementation.

V. Numerical Case Study

Here, we numerically evaluate the optimistic elimination framework on synthetic problems and an engineering co-design example, with the goal of quantifying sample-efficiency gains from structural optimism under a common evaluation budget. We begin by describing the reference algorithms and the adaptations needed for a fair comparison.

A. Benchmark Algorithms and Performance Metric

We compare our method against four EAs: MOEA/D [8], which decomposes the multi-objective problem into cooperatively solved scalar subproblems; NSGA-III [7], which combines non-dominated sorting with reference-point-based diversity preservation; RVEA [19], which steers the population via angle-penalized reference vectors that adapt to irregular front shapes; and KGB [20], which uses a naive Bayesian classifier on past solutions to seed a high-quality initial population. Moreover, we compare with qLogEHVI [21], a Bayesian optimization method with expected hypervolume improvement acquisition function whose log formulation improves numerical stability. Finally, we compare against a structure-agnostic quasi-random sampler based on the Halton sequence [22]. This same Halton sequence serves as the base sampling measure μ used in Algorithm 1.

All baselines operate on the same local implementation space of the expensive oracle, ensuring a uniform notion of sample complexity[1]. Two adaptations ensure fair comparison: (i) performance is measured from the cumulative archive of all unique evaluations up to budget t, not just the final population of EAs; and (ii) implementations failing to meet the target functionality f are assigned the worst resource value $\top_{\mathcal{R}}$, converting infeasibility into domination for algorithms that do not reason about target feasibility.

In all examples, $\mathcal{R} = \mathbb{R}^m$ with the componentwise order. We measure approximation quality via the *hypervolume difference*. Given a reference point $r^{\mathrm{ref}} \in \mathbb{R}^m$, the hypervolume indicator $\mathsf{HV}(A)$ is the Lebesgue volume of the region dominated by an antichain A within the box defined by r^{ref} [23]. This indicator is strictly Pareto-compliant: if one antichain dominates another, it has strictly larger hypervolume. We define $\mathsf{HVD}(A_f, \hat{A}_f) = \mathsf{HV}(A_f) - \mathsf{HV}(\hat{A}_f) \geq 0$, which is zero if and only if $\hat{A}_f$ dominates the same region as A_f (up to measure zero) and grows as the approximation worsens.

B. Lipschitz Synthetic Problems

We first evaluate the algorithms on analytically generated problems. To generate instances that adhere to the Lipschitz structure described in Section IV-C.2, we draw a random matrix $A \in \mathbb{R}^{m \times d}$ with spectral norm $\|A\|_2 = L$ and a random phase vector $b \in \mathbb{R}^m$, and define a resource map $g\colon [0,1]^d \to [0,1]^m$ by $g_j(x) = \phi\big([Ax + b]_j\big)$, $j \in \{1, \ldots, m\}$, where $\phi\colon \mathbb{R} \to [0,1]$ is the unit-slope triangle wave defined by $\phi(t) = t \bmod 2$ when $t \bmod 2 \leq 1$ and $\phi(t) = 2 - (t \bmod 2)$ otherwise, satisfying $|\phi'(t)| = 1$ almost everywhere. At any non-fold point the Jacobian is $J_g(x) = D(x)A$ with $D(x) = \mathrm{diag}(\pm 1)$ orthogonal, so $\|J_g(x)\|_2 = \|A\|_2 = L$ almost everywhere. To see that g is L-Lipschitz with respect to the ℓ_2 norm, note that ϕ is 1-Lipschitz and acts component-wise, so $\|g(x) - g(x')\|_2 \leq \|Ax - Ax'\|_2 \leq \|A\|_2\,\|x - x'\|_2 = L\,\|x - x'\|_2$, and the bound is attained along the top singular vector of A.

In Table I, we report the cumulative hypervolume difference HVD on 8 random Lipschitz problems with $L = 2$. For the cumulative HVD, a lower value is better, and the worst-case value scales linearly with the number of iterations. Each figure represents an average of 100 independent runs. As a proxy for the true antichain, we use the best antichain recovered among all runs and algorithms.

Algorithm 1 attains the best result on 5 out of 8 instances and the second-best result on 2 of the remaining 3. Notably, while MOEA/D is competitive on several Lipschitz instances, it exhibits high variance across problems (e.g., strong on L2 and L5 but poor on L8), whereas Algorithm 1 delivers consistently low HVD across all instances. These results separate two effects. The improvement over the pure Halton baseline shows the value of elimination itself, beyond uniform low-discrepancy coverage of the implementation space. The additional improvement over the EAs shows the benefit of exploiting explicit structural knowledge that is unavailable to generic population-based heuristics. Overall,

[1] All experiments were run on a machine with an AMD Ryzen 9 7900X CPU (12 cores, 4.7 GHz) and 64 GB of RAM.

TABLE I

THE CUMULATIVE HYPERVOLUME DIFFERENCE HVD (↓) ON RANDOMLY GENERATED LIPSCHITZ PROBLEMS WITH $\mathcal{I} = \mathbb{R}^4$, $\mathcal{F} = \mathbb{R}^2$, AND $\mathcal{R} = \mathbb{R}^2$ AFTER 2000 ITERATIONS. THE BEST-PERFORMING METHOD IS SHOWN IN BOLD, AND THE SECOND-BEST IS UNDERLINED.

Method	L1	L2	L3	L4	L5	L6	L7	L8
OURS	$\mathbf{5.89 \times 10^1}$	1.85×10^1	$\mathbf{4.92 \times 10^1}$	$\mathbf{2.26 \times 10^1}$	$\underline{3.80 \times 10^1}$	$\underline{1.79 \times 10^1}$	$\mathbf{4.63 \times 10^1}$	$\mathbf{4.20 \times 10^1}$
HALTON	2.84×10^2	2.77×10^1	2.55×10^2	$\underline{2.52 \times 10^2}$	7.04×10^1	2.94×10^1	9.85×10^1	$\underline{2.96 \times 10^2}$
NSGA-III	1.82×10^2	2.00×10^1	1.29×10^2	3.54×10^2	4.93×10^1	2.12×10^1	6.37×10^1	8.21×10^2
MOEA/D	3.13×10^2	$\mathbf{1.39 \times 10^1}$	1.25×10^2	3.87×10^2	$\mathbf{3.35 \times 10^1}$	$\mathbf{1.59 \times 10^1}$	$\underline{4.71 \times 10^1}$	1.04×10^3
RVEA	$\underline{1.79 \times 10^2}$	$\underline{1.76 \times 10^1}$	$\underline{1.18 \times 10^2}$	3.34×10^2	4.46×10^1	2.03×10^1	5.66×10^1	1.01×10^3
KGB	2.37×10^2	2.02×10^1	1.55×10^2	4.39×10^2	4.95×10^1	2.17×10^1	7.04×10^1	9.57×10^2

the synthetic experiments support the central claim of the paper: when valid optimistic certificates can be constructed, they translate directly into improved sample efficiency.

C. Intermodal Mobility System Co-Design

We next consider a realistic problem of intermodal mobility system co-design, following the model of [2], [24]. A central planner jointly selects Autonomous Vehicle (AV) characteristics and public-transit service levels in the context of an Autonomous Mobility-on-Demand (AMoD) service, seeking to serve a prescribed travel demand while trading off average travel time against total system cost. The transportation network is modeled as an edge-labeled digraph $G = \langle V, A, c \rangle$ with four layers—road networks for AVs and micromobility, a public transit network, and a walking network—linked by mode-switching arcs. Customer movements are specified by travel requests $Q = \{\langle o_m, d_m, \alpha_m \rangle\}_{m=1}^{M}$ with origin–destination pairs and request rates, and a multi-commodity flow model routes customers and rebalancing vehicles subject to flow conservation and capacity constraints [24].

In the notation of Definition II.8, the functionality poset $\mathcal{F}_{\text{mob}}$ is the set of serviceable travel demands ordered by inclusion and rate dominance, and the resource poset is $\mathcal{R}_{\text{mob}} = \langle \mathbb{R}^2_{\geq 0}, \preceq \rangle$, representing average travel time and total cost (fleet acquisition, operations, and infrastructure for both AVs and public transit). Each implementation $i = \langle v_V, n_V, n_S \rangle$ specifies the AV speed $v_V \in [20, 50]$ mph, fleet size $n_V \in \{0, \dots, 5000\}$, and subway service frequency $n_S \in [0, 1]$. The *expensive block* v_q is the intermodal system itself: for each candidate, the resource map $\mathsf{req}(i) = \langle t_{\text{avg}}, C_{\text{tot}} \rangle$ is obtained by solving a multi-commodity flow LP that minimizes average travel time over customer and rebalancing flows subject to conservation, capacity, and fleet-size constraints. Each evaluation thus requires solving a large-scale LP, making it the computational bottleneck and the natural target for adaptive sampling via Algorithm 1.

Case Study Details: For this problem, exact computation of the optimal nondominated sets is intractable, so we discretize the implementation spaces to $|\mathcal{I}_{\text{mob}}| = 2197$ to enable cached evaluations. The EAs are tuned over population sizes $\{50, 100, 200\}$, distribution indices $\{5, 10, 15\}$, and crossover probabilities $\{0.1, 0.5, 0.9\}$. For QLOGEHVI, the Gaussian process fitting and acquisition function optimization are aware of the noiseless setup. Our method applies the relaxed monotone optimistic evaluation of Section IV-C.1

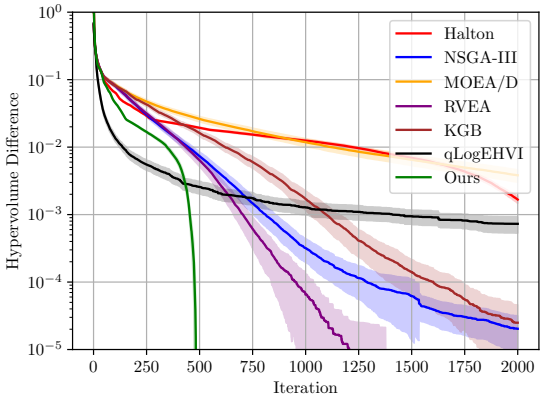


Fig. 3. Hypervolume difference HVD as a function of the number of iterations for the intermodal mobility co-design problem.

with forced acceptance probability $\delta = 0.05$. The intuition behind monotonicity is that t_{avg} likely decreases with a larger AV fleet while C_{tot} increases, and analogous reasoning applies to the remaining design variables (subway frequency and AV speed). Since monotonicity may not hold exactly, we replace the maximum in (11) by the average of its K largest elements for $K = 2$, and cap the rejection-sampling loop at 50 iterations.

Results: Figure 3 reports the HVD as a function of the number of iterations, where each curve is an average of 500 independent runs (50 for QLOGEHVI), and the shaded regions represent half the standard deviations. Our approach recovers the exact antichain substantially faster than all baselines, reaching low HVD values with far fewer evaluations of the expensive block v_q. Notably, the low-discrepancy HALTON sampler performs poorly and is outpaced by several EAs, indicating that the antichain is concentrated in a small region of the implementation space that space-filling heuristics are slow to locate. Our approach exploits the (approximate) monotone structure of the co-design problem to focus evaluations precisely on that region: it requires approximately 33% fewer samples than RVEA—the best-performing baseline—to reach an HVD of 10^{-3}, and approximately 60% fewer samples to reach 10^{-5}. In addition to improved sample efficiency, the optimistic elimination sampler exhibits lower variance than all population-based baselines. Finally, although QLOGEHVI converges faster than the other methods early on, it eventually plateaus and

TABLE II

ABLATION STUDY ON RELAXED REJECTIONS IN ALGORITHM 1. ALL METRICS ARE EVALUATED AT ITERATION $N = 2000$.

Metric	OURS ($\delta = 0, K = 1$)	OURS ($\delta = 0.05, K = 2$)	RVEA	HALTON
Cumulative HVD (↓)	18.53	19.95	27.27	41.61
Exact Recovery (↑)	89.8%	100%	99.8%	0%

TABLE III

CPU TIME PER STEP FOR ALGORITHM 1, HALTON, NSGA-III, QLOGEHVI, AND THE MOBILITY LP v_q.

Time	OURS	HALTON	NSGA-III	QLOGEHVI	Sim. v_q
Avg.	2.61×10^{-3}	0.92×10^{-3}	1.07×10^{-3}	3.88×10^{-1}	5.84×10^{1}
Min.	0.16×10^{-3}	0.15×10^{-3}	0.17×10^{-3}	1.34×10^{-1}	5.22×10^{0}
Max.	5.23×10^{-2}	2.01×10^{-3}	2.19×10^{-3}	9.57×10^{-1}	1.70×10^{2}

struggles to recover the exact antichain.

To isolate the effect of elimination under approximate monotonicity, we report performance metrics after 2000 iterations for our approach with and without relaxed rejections in Table II, where exact recovery means the true antichain of the discretized benchmark is identified up to machine precision. The results show that our approach degrades gracefully under mild model misspecification, and that relaxed rejection substantially improves recovery probability at the cost of a slight reduction in convergence speed.

Finally, Table III reports the per-step CPU time of Algorithm 1, HALTON, NSGA-III, QLOGEHVI, and the mobility LP v_q. Our method is slower than HALTON and NSGA-III due to the rejection-sampling loop, with roughly a twofold increase in average runtime and up to an order-of-magnitude increase in the worst case. Nevertheless, even its worst-case CPU time is still two orders of magnitude smaller than the minimum time needed to solve a single mobility LP and substantially lower than the Bayesian optimization baseline, making the additional overhead worthwhile given the improvement in convergence and antichain recovery.

VI. CONCLUSIONS AND FUTURE WORK

We presented an online learning framework for multi-objective co-design in expensive system design loops, where optimistic bounds safely eliminate unpromising implementations before invoking a costly solver, simulator, or experiment. The method provably preserves all designs that could improve the current Pareto set, and the search region contracts monotonically with new observations. A key feature is that the approach is generic over the order structure of functionalities and resources, leaving the modeling expressiveness of monotone co-design theory fully intact. Experiments on mobility and synthetic benchmarks show that the approach matches or improves upon leading baselines while using substantially fewer expensive evaluations.

Key open directions include developing a compositional theory for problems described by a multigraph-structured model, deriving problem-dependent sample complexity bounds, and developing more sophisticated acquisition strategies for high-dimensional design spaces.

REFERENCES

[1] M. Ehrgott, *Multicriteria Optimization*. Springer, 2005.

[2] G. Zardini, "Co-Design of Complex Systems: From Autonomy to Future Mobility Systems," Ph.D. dissertation, ETH Zurich, 2023.

[3] A. Moitra and R. O'Donnell, "Pareto Optimal Solutions for Smoothed Analysts," in *Proceedings of the forty-third annual ACM symposium on Theory of Computing*, 2011, pp. 225–234.

[4] D. Dörfler, A. Löhne, C. Schneider, and B. Weißing, "A Benson-Type Algorithm for Bounded Convex Vector Optimization Problems with Vertex Selection," *Optimization Methods and Software*, vol. 37, no. 3, pp. 1006–1026, 2022.

[5] A. Censi, "A Mathematical Theory of Co-Design," *arXiv preprint arXiv:1512.08055*, 2015.

[6] A. Censi, J. Lorand, and G. Zardini, "Applied Compositional Thinking for Engineering," 2024, work-in-progress book.

[7] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2013.

[8] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[9] K. Li, "A Survey of Multi-objective Evolutionary Algorithm Based on Decomposition: Past and Future," *IEEE Transactions on Evolutionary Computation*, 2024.

[10] J. Knowles, "ParEGO: A Hybrid Algorithm with On-line Landscape Approximation for Expensive Multiobjective Optimization Problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50–66, 2006.

[11] S. Daulton, M. Balandat, and E. Bakshy, "Differentiable Expected Hypervolume Improvement for Parallel Multi-Objective Bayesian Optimization," *Advances in neural information processing systems*, vol. 33, pp. 9851–9864, 2020.

[12] D. Hernández-Lobato, J. Hernandez-Lobato, A. Shah, and R. Adams, "Predictive Entropy Search for Multi-objective Bayesian Optimization," in *International conference on machine learning*. PMLR, 2016, pp. 1492–1501.

[13] P. Auer, C.-K. Chiang, R. Ortner, and M. Drugan, "Pareto Front Identification from Stochastic Bandit Feedback," in *Artificial intelligence and statistics*. PMLR, 2016, pp. 939–947.

[14] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge University Press, 2020.

[15] C. Ararat and C. Tekin, "Vector Optimization with Stochastic Bandit Feedback," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023, pp. 2165–2190.

[16] C. Kone, E. Kaufmann, and L. Richert, "Bandit Pareto Set Identification in a Multi-Output Linear Model," *arXiv preprint arXiv:2507.04255*, 2025.

[17] é. crepon, A. Garivier, and W. M. Koolen, "Sequential Learning of the Pareto Front for Multi-objective Bandits," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2024, pp. 3583–3591.

[18] S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvári, "X-Armed Bandits," *Journal of Machine Learning Research*, vol. 12, no. 5, 2011.

[19] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[20] Y. Ye, L. Li, Q. Lin, K.-C. Wong, J. Li, and Z. Ming, "Knowledge Guided Bayesian Classification for Dynamic Multi-Objective Optimization," *Knowledge-Based Systems*, vol. 250, p. 109173, 2022.

[21] S. Ament, S. Daulton, D. Eriksson, M. Balandat, and E. Bakshy, "Unexpected Improvements to Expected Improvement for Bayesian Optimization," *Advances in neural information processing systems*, vol. 36, pp. 20 577–20 612, 2023.

[22] J. H. Halton, "Algorithm 247: Radical-Inverse Quasi-Random Point Sequence," *Communications of the ACM*, vol. 7, no. 12, pp. 701–702, 1964.

[23] M. Li and X. Yao, "Quality Evaluation of Solution Sets in Multiobjective Optimisation: A Survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, pp. 1–38, 2019.

[24] G. Zardini, N. Lanzetti, A. Censi, E. Frazzoli, and M. Pavone, "Co-Design to Enable User-Friendly Tools to Assess the Impact of Future Mobility Solutions," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 2, pp. 827–844, 2022.