CONTINUATION-CONDITIONED CREDIT FOR COMPOSING VEHICLE ROUTING HEURISTICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Automatic heuristic design typically ranks programs by the quality of their complete solutions. Yet a program that is weaker on its own may supply decisions that improve another policy. We study this distinction through continuation-conditioned credit and *Cross-Family Residual Composition* (CFRC). CFRC uses a genetic programming (GP) scaffold and searches over bounded donor deviations evaluated under the continuation actually deployed. In an independent eight-seed capacitated vehicle routing (CVRP) campaign, changing only the continuation reverses rankings in 35.24% of 2,957 supported action contrasts. Retrospective evaluations on two independent four-pool CVRP-30 cohorts compare composition with the per-instance best complete execution of all sixteen source candidates plus the scaffold. Under a common numeric execution environment, composition yields mean paired log gains of 1.758% (95% CI $[0.215, 3.123]$) and 3.314% ($[2.594, 3.955]$). On 144/480 and 255/480 instances, respectively, every composed deviation is supported only by donors worse than the scaffold as standalone policies, yet the composition improves it. Separately, prospective comparisons matching successful generation evaluations, selector capacity, and $K = 2$ deployment find mixed-donor gains of 0.625% and 0.496% over GP-only donors. Deviation-budget and donor-count ablations show how additional inference computation improves solution quality. Together, these results demonstrate that action-level composition can improve on complete-program reuse from the same candidate pool on random CVRP, with transfer varying across distributions.

1 INTRODUCTION

Automatic heuristic design (AHD) searches for reusable decision programs. In capacitated vehicle routing (CVRP), a program builds a solution through successive customer choices. LLM- and search-based systems generate and select programs differently, yet typically assign fitness to the complete program (Romera-Paredes et al., 2024; Liu et al., 2024; van Stein & Bäck, 2025; Ye et al., 2024; Zheng et al., 2025; Novikov et al., 2025). Complete-program fitness answers which policy works alone; it does not measure which of its decisions help another policy.

The reason is continuation dependence. At a construction state, the terminal effect attributed to an action includes every decision made after it. If two programs disagree, forcing either action and completing with one program can favor a different action than completing with the other. Whole-program fitness therefore confounds an action with the continuation under which it was observed. Across independently generated CVRP programs and instances, $1{,}042/2{,}957$ supported action contrasts (35.24%) reverse sign when only the continuation changes. Reversals cover both EoH and MCTS-AHD trajectory supports, all eight frozen program pairs, and all 12 instances. This motivates separating a program's standalone quality from the value of its decisions under another policy.

Cross-Family Residual Composition (CFRC) implements this test. A GP program supplies the scaffold; independently generated programs propose feasible actions at construction states. CFRC scores each proposal with the bounded continuation it uses at deployment. The scaffold action remains available without spending the deviation budget, so the deployed solution realizes the evaluated objective, uses at most K donor deviations, and is pointwise no worse than the scaffold.

The resulting *continuation-conditioned residual value* measures an action under a specified continuation and remaining deviation budget. Rollout and limited discrepancy search supply the improve-

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

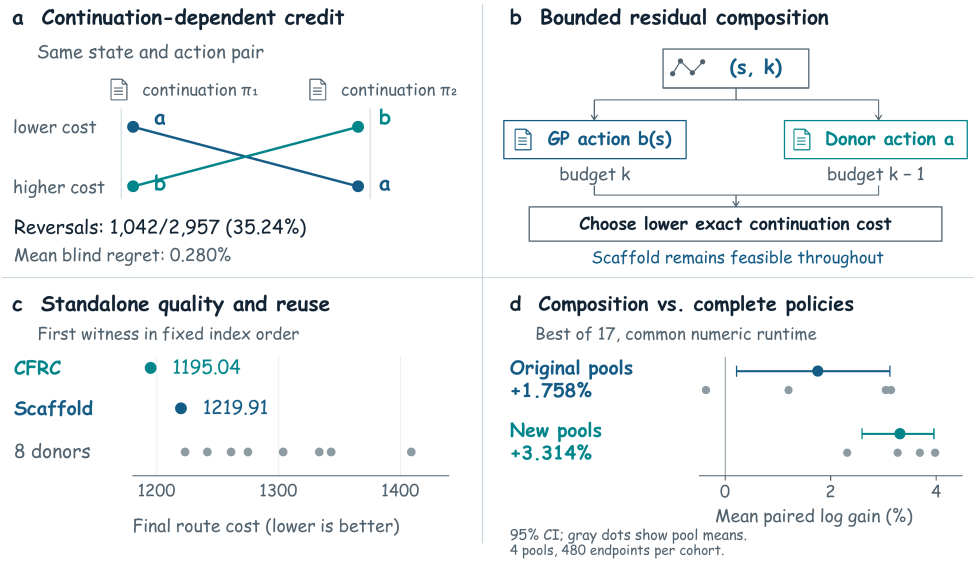

Figure 1: **Standalone program quality and continuation-conditioned reuse.** (a) Schematic ranking reversal with measured CVRP frequency and blind regret. (b) Bounded composition retains scaffold actions at no deviation cost. (c) First archived, independently replayed witness in pool/panel/instance order. All eight donors shown are worse standalone policies; the numeric-runtime rerun reproduces this endpoint. (d) Composition versus best-of-seventeen under a common numeric runtime, on four pools and 480 endpoints per cohort. Bars show hierarchical 95% intervals; gray dots show every pool mean. These retrospective evaluations reuse the original sources and endpoints (Table 4); inference time is not matched.

ment principle and bounded neighborhood (Bertsekas et al., 1997; Harvey & Ginsberg, 1995). We use this neighborhood to measure what standalone fitness omits. On 144/480 and 255/480 instances in two cohorts, every composed deviation comes only from donors worse than the scaffold as complete policies on that instance, yet their composition improves it. Standalone superiority is therefore not necessary for useful reuse.

Two comparisons distinguish action composition from proposal-family choice. First, retrospective evaluations retain the source programs and endpoints, using a common numeric runtime from donor selection through deployment. Composition beats per-instance best-of-seventeen by 1.758% and 3.314% in the original and new CVRP-30 cohorts. Second, prospectively fixed comparisons hold the scaffold, selection capacity and deployment rule constant: mixed proposals improve over GP-only by 0.625% ($[0.189, 1.186]$), replicated at 0.496% ($[0.069, 0.852]$) in four new pools. Planning and action-set interventions probe separate sources of endpoint gain; scale, donor-count and transfer tests locate the quality–cost tradeoff and its distributional limits.

The paper makes three contributions:

- a CVRP action-credit intervention that exposes continuation-dependent ranking reversals and quantifies the cost of continuation-blind ordering;
- controlled evidence that generated programs can be useful residual proposers despite inferior standalone performance, including full-candidate complete-policy controls and deployed-action witnesses; and
- a bounded composition study separating proposal-family gains from planning, with two independent four-pool confirmations and quality–cost and transfer measurements.

2 Related Work

Automatic heuristic design. Classical hyper-heuristics search for reusable decision rules instead of direct solutions (Burke et al., 2013). LLM-driven AHD couples program proposals with evolutionary selection and memory through islands, idea–code evolution, reflective prompts, or program databases (Romera-Paredes et al., 2024; Liu et al., 2024; van Stein & Bäck, 2025; Ye et al., 2024; Novikov et al., 2025). Recent systems refine operator choice, search memory, path-level credit, and code editing (Huang et al., 2026; Liu et al., 2026b; Zhao et al., 2026a; Chen et al., 2026; Qiu et al., 2026; Ahmed et al., 2026; Liu et al., 2026a; Gungordu et al., 2026; Wang et al., 2026; Lai et al., 2026; Zhao et al., 2026b; Sun & Shi, 2026). Our question concerns the value of a generated program's decisions under a specified continuation. Appendix C details the deployment contracts of peer systems.

Tree search, GP, and adaptive selection. MCTS-AHD organizes LLM-generated heuristics in a search tree to explore successive refinements (Zheng et al., 2025). GP evolves program populations through mutation and crossover (Koza, 1992); context- and impact-aware crossover evaluates alternative offspring during generation (Majeed & Ryan, 2006; Majeed et al., 2021). Adaptive use of evolved heuristic pools is also established. HeurAgenix uses single-operation counterfactuals during evolution and combines LLM shortlisting with Monte Carlo completion for online selection (Yang et al., 2025); DSevolve selects a scheduling rule from state features, with optional short look-ahead (Zhou et al., 2026). These systems already condition heuristic choice on the current state and future completion. Our comparison object is the opportunity loss from judging independently generated programs only by complete-policy cost: fixed-source composition, weaker-donor witnesses, and joint-pruning bounds measure useful decisions omitted by that ranking. A separate comparison changes proposal family under the same composition rule.

Component context and credit. Cooperative coevolution evaluates coadapted subcomponents through collaborators (Potter & De Jong, 2000); collaborator choice can change the credit assigned to a component (Wiegand et al., 2001), and decomposed evaluation functions can reduce whole-solution confounding when a useful factorization is available (Agogino et al., 2005). Semantic and behavioral GP likewise extract more feedback than a scalar program fitness (Moraglio et al., 2012; Krawiec & O'Reilly, 2014). Recent AHD systems also coordinate coupled subproblems, coevolve component-specific operator populations, evolve multi-heuristic ensembles, or jointly evolve interacting routing components (Bömer et al., 2026; Kuang et al., 2026; Lv et al., 2026; Wei et al., 2026). Those systems optimize coordinated tuples or predefined functional components. Here the components are decisions proposed by independently generated complete programs; their value is measured under a shared continuation.

3 Mechanism Evidence and Evaluation Protocol

3.1 Continuation-dependent action credit

For a decision state x, feasible action a, and complete continuation policy π, let $L(x, a; \pi)$ be the terminal loss obtained by forcing a once and then following π, including the domain's unchanged fixed post-processing. For two actions a and b, define

$$\Delta_\pi(x; a, b) = L(x, a; \pi) - L(x, b; \pi). \tag{1}$$

For minimization, $\Delta_\pi > 0$ favors b. We call $\Delta_\pi \Delta_{\pi'} < 0$ a *strict continuation reversal*: the same action pair changes order when only the completion program changes. This intervention supplies the mechanism diagnostic behind CFRC; it is not itself an algorithm comparison.

We freeze native EoH and MCTS-AHD CVRP programs paired by search seed, enumerate non-equivalent action disagreements on both trajectory supports, and evaluate the four action–continuation combinations. The discovery protocol is repeated unchanged on independent instances and again in an independent program campaign. All policy–instance endpoints must exactly reproduce the native drivers.

3.2 PRIMARY COMPARISON AND REPORTING

The primary scientific estimand is the change in a CVRP action contrast when only its continuation changes. The primary system test is the prospectively frozen mixed-family versus GP-only residual-set confirmation. Both systems account for 960 successful generation evaluations, share the scaffold, expose sixteen candidates, select eight donors, deploy nine programs, and use exact $K{=}2$ on identical endpoints. The test contains four independent pools, five panels per pool, and 24 instances per panel. We report a 20,000-draw hierarchical bootstrap over pools, panels, and instances, alongside pool/panel wins, evaluated/deployed equality, pointwise scaffold safety, deviation-cap compliance, and median deployment time. API attempts/tokens, extraction, selection compute, and wall time are separate resource axes. An orthogonal JSSP intervention and a BinPack identification check remain secondary to the CVRP claim. Appendix A records the scope decision and tabulates the generation and deployment contracts of the system controls.

4 CONTINUATION-DEPENDENT CREDIT

Table 1 reports the CVRP intervention in Section 3.1. In the discovery panel, 193 of 600 supported contrasts (32.17%) reverse sign; independent instances yield 210 of 618 reversals (33.98%). Reversals occur on both EoH and MCTS-AHD trajectory supports, in every retained deterministic program pair, and in all six instances of each panel. A separate eight-seed campaign changes both programs and instances and reproduces the effect: 1,042 of 2,957 contrasts (35.24%) reverse. They cover both supports, all eight frozen program pairs, and all 12 instances. The native policy–instance endpoints exactly reproduce their unchanged drivers.

Panel	Pairs	Instances	Contrasts	Reversals	Rate
Discovery	4	6	600	193	32.17%
Independent instances	4	6	618	210	33.98%
Independent programs and instances	8	12	2,957	1,042	35.24%

Table 1: CVRP four-way action/continuation intervention. The independent campaign changes both programs and instances without quality selection; strict sign reversals are reported.

An orthogonal JSSP intervention yields 106 reversals among 641 deviations from 12 generated programs. Its action-pair and continuation construction differs, so its rate is not pooled with CVRP (Appendix C).

Proposition 1 (Continuation-blind regret)**.** *Let continuations π_1, π_2 occur with probabilities $p, 1{-}p$, and write $\delta_j = L(x, a; \pi_j) - L(x, b; \pi_j)$. Relative to a selector that observes the continuation, the minimum expected excess loss of any continuation-blind, possibly randomized selector is*

$$\min\{p[\delta_1]_+ + (1-p)[\delta_2]_+,\ p[-\delta_1]_+ + (1-p)[-\delta_2]_+\}\,. \tag{2}$$

Under a strict reversal this equals $\min\{p|\delta_1|, (1{-}p)|\delta_2|\}$; if $p \in [\alpha, 1{-}\alpha]$, it is at least $\alpha \min_j |\delta_j|$. Appendix B gives the proof.

A post-hoc diagnostic sets $p = 1/2$ and normalizes the row-wise regret by the mean of the four terminal costs. On the independent CVRP panel, its mean over *all* supported contrasts is 0.280% (cluster-bootstrap 95% CI $[0.232, 0.333]$); non-reversals contribute zero. Conditional on reversal, the median is 0.536%. The discovery value has the same sign. Even when both continuations have probability at least 0.25, the all-contrast lower bound remains 0.140%. Thus, on the measured support, hiding the continuation has a nonzero cost for an oracle blind ordering under this hypothetical mixture.

A separate deployment intervention conditions on the continuation explicitly. Table 2 reports untouched endpoint panels. Bounded $K{=}2$ improves every frozen CVRP pair, with a 2.735% log effect over the base; the orthogonal JSSP control improves 2.174% over its frozen base and 1.700% over the better complete parent. Recursive $K{=}2$ improves all four CVRP pairs by 2.424% over the base, while every instance independently reproduces its evaluated endpoint and is non-inferior to the base. The median CVRP runtime ratio is 15.39× in this control.

Consequence	CVRP	JSSP
Blind regret, all contrasts (%)	0.280	—
Blind regret, reversal median (%)	0.536	—
Bounded K=2 vs. base (log-%)	+2.735	+2.174
Recursive K=2 vs. base (log-%)	+2.424	—
Fresh program-pair wins	4/4	4/4

Table 2: Consequences of continuation dependence on separately frozen panels. Blind regret uses $p = 1/2$ in Equation 2, normalized per contrast. Bounded and recursive K=2 are classical rollout controls using the frozen AHD proposal pair, not new search algorithms. JSSP has no recursive confirmation.

BinPack supplies no non-equivalent action support: frozen pairs choose the same bin or bins with equal remaining capacity (Appendix C).

5 Continuation-Conditioned Residual Composition

5.1 Residual value under the deployed continuation

Let b be a deterministic scaffold and $\mathcal{D} = \{d_1, \ldots, d_m\}$ deterministic donor heuristics. At construction state s, each policy proposes a feasible action and $T(s, a)$ is the successor state. Rather than rank donors by standalone terminal cost, we ask whether a donor action improves the objective when subsequent decisions follow the same deployed continuation.

For a minimization problem and deviation budget $k > 0$, CFRC computes

$$V_k(s) = \min\left\{ V_k(T(s, b(s))) \,, \min_{a \in A_{\mathcal{D}}(s) \setminus \{b(s)\}} V_{k-1}(T(s, a)) \right\}, \tag{3}$$

where $A_{\mathcal{D}}(s) = \{d(s) : d \in \mathcal{D}\}$ after equivalent actions are merged. At a terminal state, $V_k(s)$ is its final cost, including the fixed post-processing, for every k. At $k = 0$, $V_0(s)$ completes with b. Following b preserves the budget; taking a non-equivalent donor action consumes one unit. Thus k counts departures from the scaffold rather than construction depth: the base action is free, $k = 0$ exactly reproduces the scaffold, and the scaffold remains a nested safety control. The main comparisons use $K = 2$; a frozen CVRP sensitivity panel evaluates $K \in \{0, 1, 2\}$. Memoization keys the serialized state and remaining budget. The construction horizon decreases on every transition, so the recurrence is acyclic even when the scaffold branch preserves k. If the horizon is H and at most r merged non-base actions are proposed per state, exact enumeration has at most $\sum_{j=0}^{K} \binom{H}{j} r^j$ terminal trajectories before memoization; fixed small K is therefore essential rather than incidental.

Proposition 2. ***Scaffold safety and exact deployment.*** *For every instance and $K \geq 0$,* CFRC *returns a feasible solution whose cost is no greater than completing with b. Following minimizing branches has exactly value $V_K(s_0)$ and makes at most K non-base decisions.*

Proof. Induct on the remaining construction horizon. The scaffold branch is always feasible, so $V_k(s)$ is no larger than following b from s. Each donor branch advances the state and decreases k. Selecting a minimum with fixed tie-breaking therefore realizes the recurrence to a terminal state, and no trajectory can contain more than K donor branches. □

The GP scaffold supplies a complete policy; independently generated programs supply residual proposals. CFRC combines the policy-improvement view of rollout (Bertsekas et al., 1997) with a discrepancy-bounded search neighborhood (Harvey & Ginsberg, 1995); donor programs define the non-scaffold action set. This makes their actions measurable under another policy's continuation.

Training-selected deployment commits to one complete policy; per-instance best-of-program deployment executes several policies and retains their best solution. Voting instead combines immediate donor actions. CFRC evaluates actions proposed by independently synthesized programs with their deployed continuation and limits the trajectory to K scaffold departures. The comparison with GP-only residual sets tests whether the generated proposal family matters under this shared deployment rule.

5.2 COMPOSITION AND PROPOSAL-FAMILY COMPARISONS

The primary system test compares mixed-family and GP-only residual sets. Both arms share the same GP-320 scaffold, account for 960 successful generation evaluations, expose sixteen candidates to the same frozen selector, select eight donors, deploy nine programs, and use exact $K = 2$ on identical endpoints. The primary routing instances use the unchanged random CVRP generator with $n = 30$ customers. On eight development instances per pool, the selector collects states along scaffold trajectories and greedily adds the candidate with the largest mean reduction in the exact one-deviation completion-cost envelope; hash order breaks ties. Selection is frozen before endpoint evaluation. Resource boundaries follow Section 3 and Table 9. Each cohort contains four independently generated pools, five panels per pool, and 24 paired instances per panel. Intervals use a 20,000-draw hierarchical bootstrap over pools, panels, and instances.

Program pools	Effect ↑	Hierarchical 95% CI	Pool wins	Panel wins
Original four	+0.625%	[+0.189, +1.186]	4/4	19/20
Four newly generated	+0.496%	[+0.069, +0.852]	3/4	16/20

Table 3: **Independent CVRP-30 system confirmations.** Positive paired log effects favor mixed over GP-only proposals. Both arms account for 960 successful generation evaluations, select eight donors from sixteen candidates, and deploy the same scaffold with exact $K = 2$. Each row has 480 paired endpoints; program pools and endpoints are disjoint across rows. The cohorts are analyzed separately.

The original confirmation improves objective by 0.625%, with a positive hierarchical interval, $4/4$ positive pools, and $19/20$ positive panels (Table 3). A second comparison with four newly generated pools and fresh endpoints yields $+0.496\%$ ($[0.069, 0.852]$), with $3/4$ positive pools and $16/20$ positive panels. Both protocols were frozen before their respective endpoint evaluations. Holding the scaffold, selector capacity, and exact recurrence fixed makes this a comparison of finite proposal sets. Pool-level t intervals are $[-0.291, 1.541]\%$ and $[-0.214, 1.206]\%$: with four independent pools per cohort, interval exclusion of zero is sensitive to the uncertainty model (Appendix A.5). Earlier configurations appear in Appendix C.

Both cohorts pass replay, feasibility, scaffold safety, and the $K = 2$ cap. In the original cohort, median deployment times are $6.37/3.18$ seconds for mixed/GP-only, with a $1.92\times$ median within-instance ratio. The new cohort has a $2.15\times$ median paired ratio.

Composition versus complete-program reuse. Best-of-seventeen executes the scaffold and all sixteen candidates independently, retaining its best complete solution per instance. It uses the same action adapter and route-local post-processing as CFRC. The retrospective comparison in Table 4 reuses both cohorts' sources and endpoints, with numeric support extended consistently to selection, composition and complete-policy execution. This changes six mixed donor sets but none of the GP-only sets. Re-executing both arms preserves positive mixed/GP effects: 0.642% ($[0.221, 1.193]$) and 0.527% ($[0.091, 0.869]$). Appendix A.5 records the runtime changes and all contrasts.

Table 4 shows positive mean gains over complete-policy reuse. Best-of-seventeen bounds every complete-policy subset selector on these candidates. Against this stronger control, original-cohort gains are positive in $3/4$ pools and $16/20$ panels; the new cohort has $4/4$ and $20/20$. CFRC loses on 116/480 and 44/480 instances, respectively: bounded composition and complete-policy execution explore different solution sets. The control tests reuse mode, not equal inference time. Appendix A.4 gives the earlier best-of-nine and voting controls.

Complete-policy costs and deployment traces yield 144/480 and 255/480 improving trajectories whose every deviation is supported only by donors worse than the scaffold on that instance; 133 and 238 also beat best-of-seventeen. Defining weakness by mean cost on other panels, a first-divergence bound certifies that jointly removing weak donors raises the minimum attainable $K = 2$ cost on 52/480 and 228/480 instances even when at least one donor remains. The bound uses saved exact continuation values (Appendices A.5 and B.3).

Program pools	Best-of-seventeen cost	CFRC cost	Gain [95% CI]
Original four	1022.337	1005.001	$+1.758\ [0.215, 3.123]\%$
Four new	1015.514	982.709	$+3.314\ [2.594, 3.955]\%$

Table 4: **Action composition versus complete-policy reuse.** Best-of-seventeen includes the scaffold and all pre-selection candidates. CFRC selects eight donors and deploys nine policies with $K = 2$; both use the expanded numeric runtime. Costs are arithmetic means; gains are mean paired log effects with 20,000-draw hierarchical 95% intervals. Each retrospective row reuses its cohort's 480 confirmation endpoints.

5.3 Endpoint value of planning and complementary actions

We isolate planning and proposal-set contributions on 32 fresh CVRP-30 instances, eight per frozen pool. All arms retain the same scaffold, donor programs and two-deviation cap. For each donor set, a greedy control evaluates every offered action by completing its successor with the scaffold (V_0), then repeats this choice at the next visited state. The exact policy instead plans subsequent donor deviations. A second intervention keeps exact planning but restricts available actions at every state to those proposed by both the mixed and GP-only sets, including the scaffold action. Comparing either full set with this shared-action control measures the contribution of actions unique to that set. The common control queries both sets and is a diagnostic, not a proposed deployment system.

Treatment vs. control	Effect ↑	Hierarchical 97.5% CI	Pools	Instances $+/=/-$
Mixed exact vs. mixed greedy	+3.563%	[+1.883, +5.569]	4/4	31/1/0
GP-only exact vs. GP-only greedy	+3.756%	[+2.048, +5.806]	4/4	31/1/0
Mixed exact vs. shared actions	+0.827%	[+0.148, +1.941]	4/4	20/12/0
GP-only exact vs. shared actions	+0.223%	[+0.051, +0.449]	4/4	11/21/0
Mixed exact vs. GP-only exact	+0.605%	[-0.099, +1.759]	3/4	16/9/7

Table 5: **Frozen CVRP-30 endpoint interventions.** Effects are $100 \log(\text{cost}_{\text{control}}/\text{cost}_{\text{treatment}})$ on four pools and 32 new paired instances. Intervals resample pools and instances, with 97.5% coverage for the two prespecified main contrasts. The shared-action arm applies exact $K = 2$ to the statewise intersection of the two proposal sets.

Table 5 shows a 3.563% gain from exact planning with mixed donors and a 3.756% gain with GP-only donors. Exact planning includes the greedy trajectory in its feasible search space; the comparison measures the realized quality gain alongside much greater search cost. Removing mixed-only actions loses 0.827% in final quality, while removing GP-only actions loses 0.223%. On 960 common scaffold states, mixed and GP-only each offer an action better than the best shared action under the same shared-policy continuation at 171 and 75 states, respectively. These statewise measurements characterize the offered actions. The reversal diagnostic uses separate native-program trajectories; neither test compares the residual selector with standalone-fitness donor selection.

On this smaller intervention panel, mixed versus GP-only averages $+0.605\%$, but its adjusted interval $[-0.099, 1.759]\%$ crosses zero. The two deletion effects differ by 0.605% on the log scale by construction. Median deployment times are 4.82 and 2.25 seconds for exact mixed and GP-only, versus 0.12 and 0.06 seconds for their greedy controls.

5.4 CVRP scale transfer and deviation-budget sensitivity

A sensitivity test reuses the four scaffolds and selected donor sets at $K = 0, 1, 2$ on the same 144 CVRP-30 instances. Separate transfer panels test $K = 1, 2$ at 50 and 100 customers without regenerating programs. All six mixed-versus-GP-only settings in Figure 2 have positive means, hierarchical 95% lower bounds, and four positive pool means. At the larger sizes, the two budgets use different instances, so these tests establish separate operating points rather than a paired estimate of the best

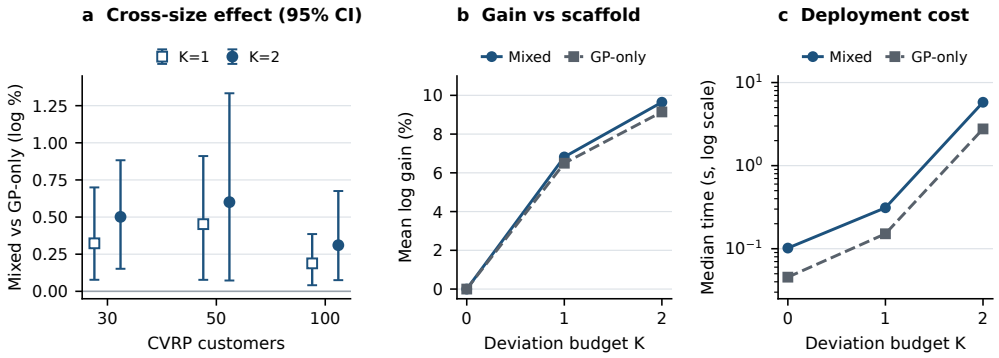

Figure 2: **Frozen CVRP hyperparameter and scale analysis.** (a) Mixed-versus-GP-only mean paired log effect, $100\log(\text{cost}_{\text{GP}}/\text{cost}_{\text{mixed}})$, with hierarchical 95% intervals over four pools and 144 endpoints per setting; positive favors mixed proposals. At $n = 50, 100$, the $K = 1$ and $K = 2$ panels use different instances. (b) Mean log improvement over the shared scaffold and (c) median per-instance deployment time on the *same* CVRP-30 endpoints for $K = 0, 1, 2$. The additional quality at $K = 2$ has a substantial exact-search cost for both donor sets. Full counts and suffix-evaluation ratios are in Table 12.

K. Replay, feasibility, scaffold safety, deviation-cap, and source-hash checks pass. Table 12 reports full counts and resource ratios.

Increasing K or a nested donor prefix expands the feasible trajectory set, so the optimal objective cannot worsen. The ablations measure the size and computational cost of that structural benefit on fixed instances. Relative to the shared scaffold, mixed composition improves by 6.820% at $K = 1$ and 9.645% at $K = 2$, while its median suffix evaluations rise from 104.5 to 3,938.5 and median deployment time from 0.31 to 5.79 seconds. GP-only composition follows the same quality–cost pattern (Figure 2; Table 11). Thus $K = 1$ provides a lower-cost operating point, while the stronger $K = 2$ endpoint requires substantially more exact continuation search. The positive mixed-versus-GP-only effect is present at both budgets across the tested CVRP sizes. At 50/100 customers, $K = 2$ mixed deployment takes median 51.4/1,255.4 seconds per instance (GP-only: 21.5/439.9 seconds); the separate $K = 1$ panels take 1.62/16.30 seconds (GP-only: 0.67/5.59 seconds). The cross-K times come from different endpoint panels and do not form a paired speedup estimate.

Donor-set capacity. A separate test varies nested donor prefixes $m \in \{1, 2, 4, 8\}$ at $K = 2$, holding generated candidates and scaffolds fixed. On 144 fresh CVRP-30 instances, expanding mixed donors from four to eight improves objective by 0.247% ($[0.101, 0.427]$), positive in all four pools and 12 panels. Median suffix evaluations rise from 2,472 to 4,067.5, and measured deployment time from 14.32 to 31.34 seconds. At eight donors, mixed improves over GP-only by 0.470% ($[0.106, 0.954]$). Table 6 gives the quality–time curve: more donors improve quality at additional inference cost, without changing generation cost.

m	Mixed gain	GP-only gain	Mixed vs. GP-only [95% CI]	Mixed time	GP-only time
1	7.708%	7.513%	+0.195% [-0.312, +0.711]	1.41 s	1.90 s
2	8.904%	8.677%	+0.227% [-0.256, +0.724]	4.41 s	4.57 s
4	9.741%	9.342%	+0.398% [-0.113, +0.967]	14.32 s	9.53 s
8	9.988%	9.518%	+0.470% [+0.106, +0.954]	31.34 s	13.72 s

Table 6: **Frozen CVRP-30 donor-count ablation at $K = 2$.** Each row uses the same 144 paired instances across four program pools. Gains are mean log-percent reductions from the shared scaffold; mixed-versus-GP-only intervals hierarchically resample pools, panels, and instances. Times are armwise medians from a four-worker local run and depend on host load.

5.5 TRANSFER TO NAMED CVRP INSTANCES

A transfer test crosses 24 distinct named instances (105–199 customers) with the four program pools, retaining the shared scaffold and exact $K = 1$. Mixed donors improve quality over GP8 and its 16-donor superset GP16; the fixed-rule basis interval crosses zero, so the three-control gate fails (Table 7). Mixed deployment takes more time than each control. The parser retains coordinates, demands, and capacity, but not route-duration or service-time constraints; these are capacity-only models.

A second frozen test crosses four newly generated pools with 20 disjoint source filenames under the same $K = 1$ arms. These files yield 17 unique parsed capacity-only models. The mixed set does not reproduce its GP8 or GP16 gains, and the fixed-rule basis has lower mean cost in all four pools. Both the prespecified 20-name transfer gate and the pre-endpoint 17-model sensitivity fail (Table 16).

	Original pools: 24 models			New pools: 20 names		
Control	Gain (%)	95% CI	Pools	Gain (%)	95% CI	Pools
GP8	$+0.558$	$[+0.153, +1.095]$	$4/4$	-0.004	$[-0.405, +0.387]$	$2/4$
GP16	$+0.550$	$[+0.147, +1.082]$	$4/4$	-0.098	$[-0.557, +0.353]$	$1/4$
basis8	$+0.224$	$[-0.125, +0.771]$	$3/4$	-0.443	$[-1.115, +0.102]$	$0/4$

Table 7: **Named-CVRP transfer in both cohorts.** Positive mean paired log gain favors mixed8. Crossed 95% intervals resample four pools and 24 distinct models or 20 filenames, respectively (20,000 draws). The new cohort has 17 unique capacity-only models; its deduplicated analysis is in Table 16. Neither joint gate passes.

5.6 BOUNDED EXTERNAL CHECKS AND NEGATIVE INTERVENTIONS

Restricting each state to one non-scaffold alternative retains a positive CVRP-30 effect ($+0.545\%$, $[0.017, 1.111]$), with a $1.012\times$ paired median suffix-evaluation ratio. Its CVRP-50/100 intervals cross zero. A shared 30-second CVRP-50 deadline yields $+0.203\%$ ($[-0.460, 0.884]$) against GP16 and -0.300% ($[-0.845, 0.154]$) against the fixed-rule basis. Appendix C gives these controls, the unsuccessful family-allocation test, and the distinct deployment contracts of native AHD peers.

6 DISCUSSION, LIMITATIONS, AND CONCLUSION

Standalone program quality does not fully characterize a program's value for reuse. The CVRP intervention shows that action rankings depend on the continuation, and CFRC evaluates residual proposals under the continuation it deploys. On two independent program-pool cohorts, composition improves mean solution quality over per-instance best-of-seventeen complete-policy execution. The deployed-action witnesses and conditional joint-pruning bounds show how programs weaker than the scaffold can contribute to these gains. Separate prospective comparisons under the same composition rule find positive mixed-versus-GP-only mean gains in both cohorts; with four pools each, interval exclusion of zero remains sensitive to the uncertainty model.

The benefit is higher attainable solution quality from an existing program pool. Larger deviation budgets and nested donor sets expand the feasible composition space; the ablations quantify the resulting quality and cost. On the same CVRP-30 endpoints, mixed composition's mean log gain over the scaffold rises from 6.820% at $K = 1$ to 9.645% at $K = 2$. Its median deployment time rises from 0.31 to 5.79 seconds, making the quality–cost tradeoff explicit. The shared-deadline experiment reports the time-constrained regime (Appendix C).

The demonstrated gains concern random CVRP; transfer depends on the target distribution. The first named cohort favors mixed proposals over GP-only sets, whereas the new-pool cohort does not reproduce that gain and favors the fixed-rule basis. Broader transfer and scalable continuation evaluation remain open. The design principle is to assess generated programs both as complete solvers and as sources of decisions: a weaker standalone policy can still enlarge the useful action set of a stronger heuristic.

AI USE STATEMENT

Generative AI tools assisted literature discovery, research ideation and experiment design, software implementation and result analysis, conceptual formulation and proof drafting, figure preparation, and drafting and editing manuscript text. DeepSeek API models and a local Qwen backbone generated heuristic proposals in the reported experiments. The authors checked the reported results and take responsibility for the paper.

REPRODUCIBILITY STATEMENT

The anonymous evidence package contains frozen protocols, endpoint records, analysis and evaluator code, file hashes, and independent replay audits for the main CVRP system, independent-pool replication, same-program controls, scale, K, and endpoint-intervention results, alongside donor-count records and recomputation code. The package README maps records to claims and documents one entry point for hash verification, saved-route audits and headline-statistic recomputation, without API calls. Original records and failed gates remain included.

REFERENCES

Adrian K. Agogino, Kagan Tumer, and Risto Miikkulainen. Efficient credit assignment through evaluation function decomposition. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1309–1316. ACM, 2005. doi: 10.1145/1068009.1068221.

Cheikh Ahmed, Mahdi Mostajabdaveh, and Zirui Zhou. Latent heuristic search: Continuous optimization for automated algorithm design. In *Proceedings of the Learning and Intelligent Optimization Conference (LION)*, 2026. URL https://arxiv.org/abs/2605.17137. Also arXiv:2605.17137.

Dimitri P. Bertsekas, John N. Tsitsiklis, and Cynara Wu. Rollout algorithms for combinatorial optimization. *Journal of Heuristics*, 3(3):245–262, 1997. doi: 10.1023/A:1009635226865.

Thomas Bömer, Bastian Amberg, Max Disselnmeyer, and Anne Meyer. CoupleEvo: Evolving heuristics for coupled optimization problems using large language models. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '26 Companion)*, pp. 1353–1361, 2026. doi: 10.1145/3795101.3814664.

Edmund K. Burke, Michel Gendreau, Matthew Hyde, Graham Kendall, Gabriela Ochoa, Ender Özcan, and Rong Qu. Hyper-heuristics: a survey of the state of the art. *Journal of the Operational Research Society*, 64(12):1695–1724, 2013. doi: 10.1057/jors.2013.71.

Chentong Chen, Mengyuan Zhong, Ye Fan, Jialong Shi, and Jianyong Sun. HiFo-Prompt: Prompting with hindsight and foresight for LLM-based automatic heuristic design. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=imSLzfZ6av.

Oguzhan Gungordu, Siheng Xiong, and Faramarz Fekri. MOSAIC: Adversarial co-evolution of specialist heuristics and problem instances for LLM-based automated heuristic design. *arXiv preprint arXiv:2608.07544*, 2026. URL https://arxiv.org/abs/2608.07544.

William D. Harvey and Matthew L. Ginsberg. Limited discrepancy search. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 607–613, 1995. URL https://www.ijcai.org/Proceedings/95-1/Papers/080.pdf.

Ziyao Huang, Weiwei Wu, Kui Wu, Wei-Bin Lee, and Jianping Wang. CALM: Co-evolution of algorithms and language model for automatic heuristic design. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=x6bG2Hoqdf.

John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, 1992.

Krzysztof Krawiec and Una-May O'Reilly. Behavioral programming: A broader and more detailed take on semantic GP. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 935–942. ACM, 2014. doi: 10.1145/2576768.2598288.

Mingen Kuang, Xudong Deng, Xi Lin, Ye Fan, Jianyong Sun, and Jialong Shi. LLM-driven co-evolutionary automated heuristic design for bi-component coupled combinatorial optimization. *arXiv preprint arXiv:2606.00718*, 2026. URL https://arxiv.org/abs/2606.00718.

Kezhao Lai, Yutao Lai, and Hai-Lin Liu. Beyond the node: Clade-level selection for efficient MCTS in automatic heuristic design. *arXiv preprint arXiv:2602.00549*, 2026. URL https://arxiv.org/abs/2602.00549.

Fei Liu, Xialiang Tong, Mingxuan Yuan, Xi Lin, Fu Luo, Zhenkun Wang, Zhichao Lu, and Qingfu Zhang. Evolution of heuristics: Towards efficient automatic algorithm design using large language model. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *PMLR*, pp. 32201–32223, 2024. URL https://proceedings.mlr.press/v235/liu24bs.html.

Fei Liu, Alessio Figalli, Patrick Owen, and Nicola Serra. RAISE: LLM-based automated heuristic design with robust adversary instance search. *arXiv preprint arXiv:2606.31801*, 2026a. URL https://arxiv.org/abs/2606.31801.

Fei Liu, Yilu Liu, Qingfu Zhang, Xialiang Tong, and Mingxuan Yuan. EoH-S: Evolution of heuristic set using LLMs for automated heuristic design. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 37090–37098, 2026b. doi: 10.1609/aaai.v40i43.41038.

Haoze Lv, Ning Lu, Shengcai Liu, Shaofeng Zhang, and Ke Tang. MuEvo: LLM-driven evolution of multi-heuristic ensemble. *arXiv preprint arXiv:2608.03636*, 2026. URL https://arxiv.org/abs/2608.03636.

Hammad Majeed and Conor Ryan. A less destructive, context-aware crossover operator for genetic programming. In *Genetic Programming: 9th European Conference, EuroGP 2006*, pp. 36–48. Springer, 2006. doi: 10.1007/11729976_4.

Hammad Majeed, Abdul Wali, and Mirza Beg. Optimizing genetic programming by exploiting semantic impact of sub trees. *Swarm and Evolutionary Computation*, 65:100923, 2021. doi: 10.1016/j.swevo.2021.100923.

Alberto Moraglio, Krzysztof Krawiec, and Colin G. Johnson. Geometric semantic genetic programming. In *Parallel Problem Solving from Nature – PPSN XII*, Lecture Notes in Computer Science, pp. 21–31. Springer, 2012. doi: 10.1007/978-3-642-32937-1_3.

Alexander Novikov, Ngan Vu, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025. URL https://arxiv.org/abs/2506.13131.

Mitchell A. Potter and Kenneth A. De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1–29, 2000. doi: 10.1162/106365600568086.

Zishang Qiu, Xinan Chen, Rong Qu, and Ruibin Bai. MeEvo: Metacognitive evolution combined with natural evolution for automatic heuristic design. *arXiv preprint arXiv:2606.14202*, 2026. URL https://arxiv.org/abs/2606.14202.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024. doi: 10.1038/s41586-023-06924-6.

Shengming Sun and Jialong Shi. Breaking validity-induced boundaries to expand algorithm search space: A two-stage AST-based operator for LLM-driven automated heuristic evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '26 Companion)*, pp. 1434–1440, 2026. doi: 10.1145/3795101.3814686.

Niki van Stein and Thomas Bäck. LLaMEA: A large language model evolutionary algorithm for automatically generating metaheuristics. *IEEE Transactions on Evolutionary Computation*, 29 (2):331–345, 2025. doi: 10.1109/TEVC.2024.3497793.

Yuhan Wang, Chaoda Peng, Xingyu Wu, Sheng-Hao Wu, and Zhi-Hui Zhan. How to guide LLM generation: Dual-surrogate guided search for automated heuristic design. *arXiv preprint arXiv:2607.13911*, 2026. URL https://arxiv.org/abs/2607.13911.

Juntao Wei, Yangming Zhou, Zhibin Jiang, and Shan Jiang. LLM-driven joint evolution of coupled heuristics components for routing optimization. *arXiv preprint arXiv:2609.02353*, 2026. URL https://arxiv.org/abs/2609.02353.

R. Paul Wiegand, William C. Liles, and Kenneth A. De Jong. An empirical analysis of collaboration methods in cooperative coevolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1235–1242. Morgan Kaufmann, 2001.

Xianliang Yang, Ling Zhang, Haolong Qian, Lei Song, and Jiang Bian. HeurAgenix: Leveraging LLMs for solving complex combinatorial optimization challenges. *arXiv preprint arXiv:2506.15196*, 2025. URL https://arxiv.org/abs/2506.15196.

Haoran Ye, Jiarui Wang, Zhiguang Cao, Federico Berto, Chuanbo Hua, Haeyeon Kim, Jinkyoo Park, and Guojie Song. ReEvo: Large language models as hyper-heuristics with reflective evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 43571–43608, 2024. doi: 10.52202/079017-1381.

Baoyun Zhao, He Wang, and Liang Zeng. G-LNS: Generative large neighborhood search for LLM-based automatic heuristic design. *arXiv preprint arXiv:2602.08253*, 2026a. URL https://arxiv.org/abs/2602.08253.

Jiale Zhao, Zimu Chen, Sirui Mao, Wentao Yang, Yuxiang Bai, and Liyuanjun Lai. DGA_2D: Directed graph-guided automated algorithm design with large language models. *arXiv preprint arXiv:2608.00700*, 2026b. URL https://arxiv.org/abs/2608.00700.

Zhi Zheng, Zhuoliang Xie, Zhenkun Wang, and Bryan Hooi. Monte carlo tree search for comprehensive exploration in LLM-based automatic heuristic design. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 78338–78373, 2025. URL https://proceedings.mlr.press/v267/zheng25o.html.

XinLei Zhou, Jin Huang, Jie Yang, Xinyu Li, and Liang Gao. DSevolve: Enabling real-time adaptive scheduling on dynamic flexible job shop with LLM-evolved heuristic portfolios. *arXiv preprint arXiv:2603.27628*, 2026. URL https://arxiv.org/abs/2603.27628.

A Experimental Protocols and Supporting Results

A.1 Program pools, selection, and resource accounting

Program sources. The continuation intervention uses deterministic programs already generated by native EoH and MCTS-AHD campaigns. It pairs programs by frozen search seed, enumerates states on both trajectory supports, removes equivalent actions, and evaluates the four action–continuation combinations without modifying either program. The independent campaign changes both programs and instances without quality-based filtering. CVRP contains 2,957 supported contrasts; all native policy–instance endpoints reproduce the original drivers exactly. The JSSP intervention changes only the continuation applied after an observed deviation and is reported separately because its action construction differs.

The mechanism and initial system panels use the unchanged random CVRP generator with 30 customers. The frozen transfer panel additionally evaluates 50 and 100 customers. A panel seed deterministically generates its listed number of instances; method pairs always receive identical instances.

Implementation configuration. Table 8 collects the settings recoverable from the generation scripts, frozen protocols, and manifests. The recorded API model identifier is a service alias; the archive does not pin its underlying model checkpoint. Sampling parameters omitted by the client use service defaults. The release includes the GP search implementation and a directory-restoration entry point for the numeric analysis dependencies. Exact source identities are indexed in the supplement manifest.

Component	Recorded setting
CVRP training	Six random 30-customer instances, seed 26021001 for GP; coordinates uniform on $[0, 100]^2$, demands uniform on $[5, 15]$, capacity 50.
GP search	Steady-state population 30; tournament size 3; subtree crossover probability 0.5, otherwise equally likely point or subtree mutation; failed crossover falls back to subtree mutation; worst-member replacement.
GP representation	Four features: current-to-customer and depot-to-customer distances, remaining capacity, demand. Constants $1, 0.5, 2, 0.25$; operators $+, -, \times$, protected division, min, max; generation depth 3, symbol limit 23.
GP budgets	320 unique fitness evaluations for the scaffold, 40 per donor; duplicate programs reuse cached fitness.
EoH sources	Native population size 5; best semantically unique program within the first 40 successful evaluations of each stream. Frozen protocol specifies `deepseek-flash`; new-cohort launcher sets a 2,048-token response cap and disables thinking. Temperature and top-p are not sent.
Selection	Eight development instances per pool; greedy residual envelope; eight of sixteen candidates; ascending SHA-256 tie-breaking.
Deployment	Shared capacity-feasibility adapter and fixed route-local 2-opt; exact $K = 2$ for primary CVRP-30, $K = 1$ for named transfer.
Offline verification	Python 3.12, NumPy 2.4.6; restored historical imports additionally use scikit-learn 1.7.2. No API access is needed.

Table 8: **Configuration of the primary CVRP comparisons.** Generation, development selection, and endpoint evaluation are separate stages. The per-pool search and panel seeds are preserved with the program manifests.

Four-pool replication. CVRP has four independently generated pools. A pool contains one GP-320 scaffold, eight EoH-40 programs, and eight GP-40 programs. Each EoH cell contributes its lowest-training-cost semantically unique program from the first 40 successful evaluations; the GP-40 donors come from eight independent runs. The two arms deploy the shared scaffold with their respective eight donors in frozen seed order, use exact $K = 2$, and each account for $320 + 8 \times 40 = 640$ successful generation evaluations. No development-panel donor selector is used in this replication. Five untouched panels of 24 instances per pool give 480 paired CVRP endpoints. Its hierarchical bootstrap resamples pools, panels within pool, and instances within panel.

Mixed-residual confirmation. The mixed arm receives the eight EoH-40 and eight GP-40 candidates already frozen for its pool. The GP-only arm receives sixteen GP-40 candidates, eight of which were generated prospectively after Stage A passed. Both arms share the GP-320 scaffold, expose sixteen candidates, select eight donors on the same development panel, deploy nine programs, and account for $320 + 16 \times 40 = 960$ successful generation evaluations. The mixed selections must reproduce their immutable Stage-A hashes. Selection cost is recorded separately from generation cost. On eight development instances per pool, the selector records scaffold-trajectory states, evaluates each distinct candidate action by completing with the scaffold, and greedily adds the candidate that maximally lowers the per-state cost envelope averaged equally over development instances; SHA-256 order breaks ties.

The untouched CVRP panel seeds are 66157101–66157105, 66157111–66157115, 66157121–66157125, and 66157131–66157135. Before execution, an automated repository scan found none

of these endpoint seeds outside the frozen protocol and runner. The CVRP confirmation gate requires a positive mean and hierarchical lower bound, at least three positive pool means, at least 16 positive panel means, exact mixed hash reproduction, equal accounting/capacity, and all replay and safety checks. $K = 2$ was fixed before confirmation and was not tuned on these panels. The subsequent independent K panel is described below. The CVRP confirmation pool means are $[1.429, 0.207, 0.653, 0.210]\%$. The original broader joint claim did not pass; the CVRP-specific gate did.

Scope provenance. The present CVRP focus was chosen after broader AHD exploration, whereas the CVRP endpoint protocols and their gates were recorded before their respective evaluations. We retain the original broader outcomes in the anonymous evidence archive and do not describe the later scope choice as preregistered. Method claims in this paper concern CVRP, with JSSP only an orthogonal credit check.

Resource accounting. The main controls match the scaffold, successful generation evaluations, candidate count, selected-donor count, deployed-program count, exact $K = 2$ policy, and endpoints. They do not match LLM API attempts or tokens, invalid proposal retries, source extraction, selector computation, or wall-clock cost. The 32 frozen CVRP EoH-40 donor streams contain 1,280 verified successful prefix evaluations. Their saved request and token ledgers cover the full roughly 100-sample runs, rather than the selected first-40 prefixes, so exact prefix request and token totals cannot be reconstructed from the archive. We therefore use the phrase "matched successful generation-evaluation accounting" and report inference time separately.

System	Generation account	Deployment contract
Primary mixed	GP-320 scaffold $+8$ EoH-40 $+8$ GP-40; 960 evaluations	scaffold $+8$ selected donors; exact $K = 2$
Primary GP-only	same scaffold $+16$ GP-40; 960 evaluations	scaffold $+8$ selected donors; exact $K = 2$
Four-pool CFRC / GP9-640	shared GP-320 $+8$ EoH-40 / GP-40; 640 evaluations per arm	scaffold $+8$ seed-ordered donors; exact $K = 2$
Initial CFRC	8 GP-320 $+8$ LLM-40; 2,880 evaluations	selected scaffold $+8$ LLM donors; exact $K = 2$
GP9	8 GP-320 $+8$ GP-40; 2,880 evaluations	8 GP-320 terminals $+1$ selected GP-40; exact $K = 2$
Native peers	separate EoH/MCTS-AHD/HiFo campaigns	1 selected program; native policy

Table 9: **Generation and deployment contracts.** The two primary arms share a GP scaffold, candidate capacity, selector, and endpoint panels. The four-pool replication uses fixed donor order without this selector; the separate initial CFRC–GP9 comparison uses different source-selection pipelines. Native peers are single-program operating points. API attempts, tokens, selection computation, and deployment time are separate resource axes; exact first-40 API and token totals are unavailable.

Absolute endpoint and runtime distributions. Table 10 gives the pooled absolute objectives and deployment-time distributions for the prospectively frozen mixed-versus-GP-only CVRP confirmation. It has 480 paired endpoints from four pools and 20 panels; these descriptive standard deviations and quantiles do not replace the hierarchical intervals for paired log effects in Table 3. The median within-instance mixed/GP-only CVRP time ratio is $1.92\times$, distinct from the ratio of armwise medians.

Domain	Arm	Objective mean $\pm$ SD	Wall median [10th, 90th] (s)
CVRP-30	Mixed	1005.16 ± 128.88	$6.37\ [3.15, 9.89]$
	GP-only	1011.48 ± 129.74	$3.18\ [2.07, 4.98]$

Table 10: **Absolute objectives and inference times in the frozen confirmation.** Lower is better. Statistics pool all 480 CVRP endpoints and are descriptive; inference for the paired quality contrast resamples program pools, panels, and instances as in Table 3.

A.2 Endpoint planning and action-set interventions

The frozen CVRP-30 intervention uses panel seeds 99773111–99773114, eight instances for each of the four previously selected program pools. Five arms share the original GP-320 scaffold and $K = 2$: exact mixed, exact GP-only, their statewise shared-action intersection under exact search, and a scaffold-completion greedy control for each full donor set. Greedy recomputes V_0 for every offered action at each visited state, uses the scaffold-first tie rule, and spends budget only for a non-scaffold action. All 160 endpoints pass feasibility, scaffold safety, deviation-cap, and trajectory-replay checks; exact arms also pass evaluated/deployed equality. At every scaffold-trajectory state, the common-state diagnostic values the union of offered actions under the same shared-policy V_1 continuation. The two primary contrasts are exact mixed versus greedy mixed and exact mixed versus exact GP-only; their 97.5% hierarchical intervals use 50,000 pool/instance bootstrap draws and a Bonferroni adjustment. The two-arm joint gate requires positive lower bounds and at least three positive pool means for each contrast, plus positive mixed-action deletion effects and shared-state complementarity in at least three pools. Planning and deletion conditions pass, but the mixed-versus-GP-only interval crosses zero, so the joint gate fails. The GP-only exact-versus-greedy and GP-only deletion effects are reported as symmetric controls in Table 5. The exact mixed, exact GP-only, shared, greedy mixed, and greedy GP-only median deployment times are 4.816, 2.251, 3.306, 0.122, and 0.063 seconds. These different policies use different numbers of terminal completions and search edges; the result is an objective comparison with measured inference cost, not a compute-matched inference comparison. Frozen protocol, all 32 paired rows, code hashes, source hashes, and an independent audit are retained in the supplementary evidence for this revision.

A.3 Deviation-budget, scale, and donor-count sensitivity

Each setting reuses the four confirmed scaffolds and selected donor sets on three fresh panels of 12 instances per pool (144 endpoints). CVRP-30 tests $K = 0, 1, 2$ on the same instances; CVRP-50/100 tests $K = 2$ first and $K = 1$ on separately frozen panels. For pool $p \in \{1, \ldots, 4\}$ and slot $j \in \{1, 2, 3\}$, the initial seeds are $b + 10(p - 1) + j$, where $b = 91073100, 91075100, 91077100$ for $n = 30, 50, 100$. The separate $K = 1$ transfer uses $b = 93075100, 93077100$. The two budgets at larger sizes therefore establish separate operating points, not a paired cross-K effect. Tables 11 and 12 report all results.

The 20,000-draw hierarchical analysis resamples pools, panels, and instances. A random setting requires a positive mean and 95% lower bound, at least three positive pools, and nine positive panels; the sensitivity criterion requires positive mean/lower bound and pool counts at both $K = 1, 2$. All settings pass their criteria, exact replay, feasibility, scaffold safety, deviation cap, and source-hash checks. The campaigns make no new API calls.

Residual set	K	Improvement over scaffold ↑	Median suffix evaluations	Median time (s)
Mixed	0	0.000%	0	0.102
Mixed	1	6.820%	104.5	0.311
Mixed	2	9.645%	3,938.5	5.790
GP-only	0	0.000%	0	0.045
GP-only	1	6.497%	83.0	0.151
GP-only	2	9.144%	2,308.5	2.767

Table 11: **Frozen CVRP-30 deviation-budget curve.** The shared scaffold is the $K = 0$ quality reference. Improvement is mean log-percent reduction from that reference over the same 144 endpoints; suffix evaluations and deployment times are medians.

Frozen donor-count ablation. Nested prefixes $m \in \{1, 2, 4, 8\}$ retain the selected order, GP-320 scaffold, $K = 2$, and 960-evaluation generation account. Three panels of 12 CVRP-30 instances per pool use seeds $95203100 + 10(p-1) + j$. Eight system execution orders rotate across instances, with four workers and a 180-second system timeout. All 12 panels finish; source hashes, replay, feasibility, safety, deviation caps, and pointwise prefix monotonicity pass. No new API calls occur. The primary mixed $m = 8$ versus $m = 4$ contrast has 41 strict wins and 103 ties; paired median

Random CVRP	K	Effect ↑	Hierarchical 95% CI	Pools	Panels	Suffix ratio	Time ratio
30 customers	1	+0.323%	[+0.078, +0.700]	4/4	11/12	1.18×	2.11×
30 customers	2	+0.501%	[+0.152, +0.883]	4/4	11/12	1.47×	1.95×
50 customers	1	+0.452%	[+0.077, +0.911]	4/4	10/12	1.19×	2.19×
50 customers	2	+0.601%	[+0.073, +1.334]	4/4	10/12	1.44×	2.25×
100 customers	1	+0.188%	[+0.041, +0.386]	4/4	10/12	1.17×	2.92×
100 customers	2	+0.311%	[+0.075, +0.675]	4/4	12/12	1.43×	2.10×

Table 12: **Frozen CVRP scale transfer at both deviation budgets.** Positive effect favors mixed-family over GP-only residual composition and equals $100 \log(\mathrm{cost}_{\mathrm{GP}}/\mathrm{cost}_{\mathrm{mixed}})$. Each row uses four pools and 144 paired endpoints; intervals hierarchically resample pools, panels, and instances. The CVRP-30 K values use the same endpoints, whereas the larger-size $K = 1$ and $K = 2$ transfers use different endpoint panels. Resource ratios compare mixed with GP-only.

wall and suffix-count ratios are $1.90\times$ and $1.50\times$. The $m = 2$ versus $m = 1$ and $m = 4$ versus $m = 2$ effects are 1.196% ($[0.616, 1.938]$) and 0.837% ($[0.434, 1.305]$). Table 6 gives the full curve. Wall times are from a shared host without contemporaneous records of other CPU workloads; suffix counts provide a load-independent work measure.

A.4 Complete-policy controls and archived reuse witnesses

This retrospective control reuses the original and new CVRP-30 confirmation cohorts: four program pools, five panels per pool, and 24 instances per panel in each cohort. The scaffold, eight mixed donors, endpoint seeds, and archived CFRC costs remain unchanged. Donors were selected for residual value; this control holds the pool fixed rather than optimizing a separate complete-policy ensemble. Best-of-nine executes each of the nine adapted policies to completion, applies the same route-local 2-opt, and keeps the best solution per instance. The full nine-policy cost matrix and routes are included in the artifact. Table 13 reports separate cohort means and hierarchical intervals (20,000 draws; seed 96260926). Original-cohort instance wins/ties/losses for CFRC are 386/1/93; the new cohort gives 456/0/24. All eight pool means and all forty panel means are positive. The prespecified control criterion—a positive lower interval and at least three positive pool means in both cohorts—is met.

The deployed adapter maps invalid proposals or program exceptions to the nearest feasible customer. It is unchanged across CFRC and these controls. Among 129,600 complete-policy calls per cohort, invalid proposals number 79/0 in the original/new cohorts, with no program exceptions. All 144 development policy costs reproduce native execution; 9,600 saved route sets pass independent coverage, capacity and geometric-cost checks. Scaffold costs and program hashes match the archives, and an independent bootstrap gives positive lower bounds. The artifact retains the initial adapter-mismatch failure and its pre-result correction. Local timings are not paired with archived server CFRC timings.

A secondary voting control takes the plurality action of these same nine adapted policies, preferring the scaffold action on a tied vote and then customer ID. It permits at most two scaffold departures and retains the better of the voting and scaffold solutions. CFRC gains 8.240% ($[6.524, 9.989]$) and 7.274% ($[6.466, 8.340]$) over this retained-vote control in the original and new cohorts, respectively. Best-of-nine is the primary same-program control; the easier voting contrast does not replace it.

Full-candidate complete-policy control. We next execute the scaffold and all sixteen mixed candidates before residual selection. Best-of-seventeen is the per-instance minimum over these complete solutions, so no subset selector on the same candidates can achieve a lower complete-policy cost. Under the frozen runtime, CFRC gains 1.829% ($[0.217, 3.284]$) and 3.435% ($[2.599, 4.270]$) in the original and new cohorts. Raw-call accounting exposes eight unselected LLM programs that encounter missing `isinstance` or NumPy internal-import support and use the deployment adapter's fallback. A separately recorded baseline-only sensitivity adds `isinstance` and restricts the internal import hook to NumPy and math, retaining the AST guard, programs, endpoints and CFRC outcomes. Program exceptions fall from 7,200/21,149 to zero across 244,800 complete-policy calls

per cohort; invalid proposals remain 79/0. All nine previously selected policies reproduce their original costs and routes.

Table 13 reports the baseline-only expanded-runtime effects and absolute costs. Instance wins/ties/losses are 361/1/118 and 432/0/48; original pool 3 is negative. Both runtime variants retain all 960 endpoints and independently verify 16,320 complete-policy route sets each. Intervals use the same hierarchical bootstrap as the nine-policy control. The runtime sensitivity changes only complete-policy evaluation; residual selection and CFRC outcomes remain fixed.

Program pools	Complete-policy control	Control cost	CFRC cost	Gain [95% CI]
Original four	Best-of-nine	1031.181	1005.160	$+2.593\ [1.135, 4.054]\%$
	Best-of-seventeen	1022.337	1005.160	$+1.741\ [0.211, 3.092]\%$
Four new	Best-of-nine	1027.473	982.989	$+4.447\ [3.802, 5.203]\%$
	Best-of-seventeen	1015.514	982.989	$+3.284\ [2.576, 3.928]\%$

Table 13: **Archived complete-policy controls.** These controls retain the original selected sets and CFRC endpoints. Best-of-nine uses the frozen runtime; best-of-seventeen expands numeric support for complete-policy execution only. Costs are arithmetic means and gains are mean paired log effects with hierarchical 95% intervals. Table 4 instead expands support for selection and composition as well.

Archived standalone quality and deployed reuse. A retrospective join of all 480 original-cohort complete-policy records and deployed traces finds 142 improving trajectories whose every deviation is supported only by donors that are individually worse than the scaffold on that same instance (strict cost difference $> 10^{-8}$). The four pool counts are 12, 51, 17, and 62, out of 120 each; respectively 10, 51, 14, and 60 also beat best-of-nine. Support includes every donor proposing the chosen action, not a single credited program. Using pool-mean standalone cost instead gives 0, 120, 0, and 91 such trajectories; 20 of 32 donors have higher mean cost than their scaffold. This secondary definition is not a training selector. The first instance-specific witness in pool/panel/instance order has scaffold and best-of-nine cost 1219.910 versus CFRC cost 1195.039. Independent replay reproduces all nine complete costs, the composed endpoint, and both deviation supporter sets. These observations establish reusable actions from poorer complete policies; they do not isolate individual donor effects or explain the mixed-versus-GP contrast. Raw joins and both definitions are archived.

A raw-proposal audit reconstructs all 480 deployed trajectories and probes the nine selected policies at 14,400 visited states. Of 129,600 calls per runtime, 48 return infeasible actions and none raises an exception. All 955 chosen deviations, including the 450 with LLM-only support, have at least one donor directly returning the chosen feasible action; none relies solely on fallback. This also holds for every deviation in the 142 witnesses. Expanded numeric support changes no adapted action on these states. Independent reconstruction checks feasibility, supporter sets, deviation budgets and final costs. The audit covers deployed states, including additional probes after budget exhaustion, rather than all search branches or a rerun of donor selection.

A.5 Numeric runtime from selection to deployment

We extend the same numeric support to the development selector and both composition arms, retaining all source programs, development instances, endpoint seeds, selection rules and $K = 2$. This permits `isinstance` and NumPy's internal numeric imports under the existing AST guard. A development-only audit reproduces all sixteen original ordered selections; the expanded runtime changes six of eight mixed sets and no GP-only set. Both arms then re-execute all 960 existing paired endpoints. This is a retrospective sensitivity on fixed sources, without regenerating programs or adding independent pools.

Table 14 reports every contrast, with the same 20,000-draw hierarchical bootstrap (seed 96260926). The mixed/GP criterion requires a positive mean and lower 95% bound, at least three positive pools and sixteen positive panels in each cohort. Both pass: original $4/4$ pools and $19/20$ panels; new $3/4$ and $17/20$. The changes from archived mixed costs are small and have intervals crossing zero; this is not an equivalence test. GP endpoints exactly reproduce the archive. The composition versus best-of-seventeen wins/ties/losses are 363/1/116 and 436/0/44.

Cohort	Control	Gain [95% CI] (%)	Pools	Panels
Original	GP-only	$+0.642\ [0.221, 1.193]$	4/4	19/20
	Best-of-seventeen	$+1.758\ [0.215, 3.123]$	3/4	16/20
	Archived mixed	$+0.017\ [-0.019, 0.062]$	2/4	7/20
New	GP-only	$+0.527\ [0.091, 0.869]$	3/4	17/20
	Best-of-seventeen	$+3.314\ [2.594, 3.955]$	4/4	20/20
	Archived mixed	$+0.030\ [-0.023, 0.092]$	3/4	11/20

Table 14: **Selection-to-deployment numeric-runtime sensitivity.** Positive mean paired log gain favors expanded-runtime mixed composition. Each row reuses 480 endpoints. Rows sharing a cohort are dependent contrasts, not additional replications. The final columns count positive pools and panels.

The expanded-run weak-donor join applies the same strict 10^{-8} criterion to all supporters of every deployed deviation. Table 15 separates this condition from requiring all eight donors to be worse standalone. The first witnesses in fixed index order are 1/1/8 and 5/1/0 (zero-based instance index). The original first witness retains cost 1195.039 versus scaffold cost 1219.910, as shown in Figure 1.

Cohort	Weak / instances	Beat best-nine	Beat best-seventeen	All eight worse	Raw-clean
Original	144/480	137	133	78	129
New	255/480	248	238	131	255

Table 15: **Weak-standalone donor witnesses under the common numeric runtime.** All counts concern improving compositions. "Weak" requires every supporter of every deviation to be worse standalone; the remaining columns count witnesses also satisfying the stated condition. Raw-clean requires every raw call by each supporter throughout search to be valid, a sufficient global condition rather than a test of individual deployed-event validity.

All 1,920 executed systems pass evaluated/deployed equality, feasibility, scaffold safety and the deviation cap. Independent reconstruction checks all saved states, routes and costs; a separate statistical implementation reproduces the six summaries. No raw call raises an exception. The original mixed search contains 4,466 invalid proposals handled by the unchanged adapter; the other three cohort/arm groups have none. The raw-clean witness counts are 129 and 255. Local mixed/GP median deployment times are 33.01/14.80 seconds and 29.63/13.49 seconds; these describe this run rather than a cross-machine efficiency comparison. The archived K, donor-count, planning and transfer experiments retain their original selected programs.

Average standalone quality. For each held-out 24-instance panel, we label a donor weak when its mean standalone cost on the other 96 endpoints exceeds the scaffold's by more than 10^{-8}. Under these leave-one-panel-out labels, 222/480 and 382/480 improving trajectories use only weak supporters for every deviation; 206 and 348 also beat best-of-seventeen. Full-pool labels yield 185 and 389 witnesses. All labels, pool counts and endpoint classifications are included in the supplement. These retrospective labels describe the fixed selected programs.

Joint-pruning certificate. Applying Appendix B.3 to saved exact continuation values certifies a strictly higher optimum after removing the leave-panel-out weak set on 52/480 and 228/480 endpoints with at least one donor retained. Per-pool counts are $[0, 44, 8, 0]$ and $[0, 21, 105, 102]$. The remaining original/new endpoints split into 223/12 with a surviving optimal path, 192/216 with all eight donors removed, and 13/24 with an inconclusive bound. The all-removed case reduces exactly to the scaffold and is counted separately. Assigning zero to every case outside the nontrivial certified group, the mean lower bounds over all 480 endpoints are 0.113% and 0.492% in paired log cost, after a 10^{-7} cost-unit numerical margin. In 44 and 228 certified cases, every raw call throughout the full search is valid and exception-free. The certificate concerns joint removal and assumes the recorded continuation values are exact and deterministic; it is computed from existing traces without executing a pruned policy.

Pool-level precision. A retrospective influence check removes each pool in turn, keeping cohorts separate. All remaining three-pool means are positive for the prospective mixed/GP comparison and both common-runtime contrasts above. For the prospective comparison, these means range from 0.357 to 0.764% in the original cohort and 0.366 to 0.705% in the new cohort. As an interval sensitivity, treating each pool mean as one observation gives $\bar{d} \pm t_{0.975,3} s_d/\sqrt{4}$. The resulting 95% intervals are $[-0.291, 1.541]\%$ and $[-0.214, 1.206]\%$ for prospective mixed/GP; $[-0.249, 1.532]\%$ and $[-0.182, 1.235]\%$ for common-runtime mixed/GP; and $[-0.904, 4.419]\%$ and $[2.157, 4.471]\%$ for common-runtime mixed/best-seventeen, in original/new cohort order. These intervals assume independent, identically distributed normal pool means, an assumption that four pools cannot substantiate. They supplement the frozen hierarchical bootstrap rather than replace it. The positive means withstand removal of any single pool; interval exclusion of zero is sensitive to the uncertainty model in five of six comparisons.

A.6 Named-instance and distribution transfer

Named-file transfer check. The frozen list contains 11 locally available named files with 31–101 nodes: A-n32-k5, A-n33-k5, CMT1, CMT6, CMT2, CMT7, CMT12, CMT14, CMT3, CMT8, and X-n101-k25. The measured file-weighted effect is -0.017% with crossed-bootstrap 95% CI $[-0.814, 0.648]\%$, one positive pool of four, and six positive file effects of 11; its predeclared transfer gate fails. The parser used by this check retains coordinates, demands, and capacity but does not represent route-duration or service-time fields. Consequently, CMT1/CMT6, CMT2/CMT7, CMT12/CMT14, and CMT3/CMT8 each map to identical modeled instances. The 11 files therefore yield seven distinct capacity-only CVRP instances, and the file-weighted uncertainty calculation includes these duplicates. The check characterizes this parser-defined endpoint set rather than faithful transfer to all source-file constraints.

Extended named-file transfer. A separately frozen, zero-API test uses 24 unique capacity-only named CVRP instances with 105–199 customers. Each of four independent program pools evaluates all 24 names with the same GP scaffold and exact $K = 1$ rule across mixed8, GP8, GP16, and an eight-rule fixed proposal set (basis8). This yields 96 paired rows and 384 final routes. The GP16 set contains GP8 plus eight additional GP candidates; all arms retain the scaffold action. Table 7 reports every prespecified comparator. Against GP8 and GP16, the mixed set has positive crossed-bootstrap intervals and $4/4$ positive pools. The basis8 interval crosses zero and only $12/24$ names favor mixed, so the prespecified three-control joint transfer gate fails. Mixed deployment also uses more wall time than each comparator. All 384 routes pass independent customer-coverage, capacity, objective-recomputation, exact-deployment, deviation-cap, and scaffold-safety checks; the campaign made no new API calls. As in the smaller check, the parser-defined objective retains capacity but does not add route-duration or service-time constraints from source files.

New-pool, disjoint-name transfer. Four further program pools were generated under the same 960-successful- evaluation contract and frozen before opening new endpoints. Each pool used the same selector capacity and exact $K = 2$ random-CVRP-30 deployment rule; five new panels of 24 paired instances gave the positive replication in Table 3. Mean absolute route costs are 982.989 for mixed8 and 987.987 for GP8 on these 480 paired instances. The frozen pre-endpoint gate required a positive mean and 95% lower bound, at least three positive pool means and 16 positive panel means; the observed $3/4$ pools and $16/20$ panels meet these count thresholds exactly. The generation ledger contains 1,364 HTTP attempts and 1,892,964 input-plus-output tokens across 32 native EoH runs. No generation API calls were made during either endpoint test.

The separately frozen named transfer evaluated 20 unused source filenames with the four new pools and exact $K = 1$. Source and parsed-model hashes were recorded before evaluation; three filename pairs parse to identical capacity-only models, leaving 17 unique models. The 20-name crossed bootstrap is the prespecified primary analysis, and a 17-model collapsed sensitivity was frozen before endpoint evaluation. Table 16 shows both. Neither analysis favors mixed8 over the fixed-rule basis, and the new-pool GP contrasts have intervals crossing zero. The original named joint gate and the stricter unique-model claim check both fail. All 320 evaluated routes pass customer coverage, capacity, objective recomputation, exact replay, deviation cap, and scaffold safety; GP16 contains GP8. The median paired mixed/control deployment-time ratios are $1.45\times$ against GP8,

$1.11\times$ against GP16, and $0.94\times$ against basis8. Mean absolute route costs across all pool–name pairs are 3542.138 for mixed8, 3545.050 for GP8, 3543.888 for GP16, and 3528.188 for basis8.

Control	20 names: gain [95% CI]	Pools	17 models: gain [95% CI]
GP8	-0.004 $[-0.405, +0.387]\%$	$2/4$	$+0.039$ $[-0.443, +0.503]\%$
GP16	-0.098 $[-0.557, +0.353]\%$	$1/4$	-0.024 $[-0.519, +0.481]\%$
basis8	-0.443 $[-1.115, +0.102]\%$	$0/4$	-0.281 $[-0.880, +0.297]\%$

Table 16: **Frozen new-pool named transfer.** Positive values favor mixed8. Twenty new filenames represent 17 unique parsed capacity-only models. Intervals resample four program pools and names or unique models (20,000 draws). Both analyses retain all three controls.

Clustered-coordinate transfer check. A further prospective, zero-API screen keeps the four confirmed CVRP program pools fixed and replaces uniform customer coordinates with a frozen three-Gaussian-cluster generator. It compares mixed8, GP8, and GP16 under the same exact $K = 1$ rule on 16 panels and 128 distinct paired instances across 30 and 50 customers. Against GP16, mixed8 gives $+0.285\%$ at 30 customers (hierarchical 95% CI $[-0.115, 0.711]\%$, $4/4$ positive pools) and $+0.068\%$ at 50 customers ($[-0.166, 0.362]\%$, $3/4$ positive pools). Both predeclared transfer gates fail because the intervals cross zero. Paired median deployment-time ratios are $1.174\times$ and $1.340\times$. All 128 distinct instances pass exact evaluated/deployed equality, feasibility, scaffold safety, and deviation-cap checks. This synthetic distribution shift is not a named-benchmark test or a new-method result.

B Continuation-Blind Regret and Composition Bounds

B.1 Blind-regret result

Write $\delta_j = L(x, a; \pi_j) - L(x, b; \pi_j)$. Choosing a without observing the continuation incurs $p[\delta_1]_+ + (1-p)[\delta_2]_+$ regret; choosing b incurs $p[-\delta_1]_+ + (1-p)[-\delta_2]_+$. Randomization forms convex combinations of these two values, so the minimum is their smaller endpoint. Under a strict reversal it equals $\min\{p|\delta_1|, (1-p)|\delta_2|\}$, at least $\alpha \min_j |\delta_j|$ when $p \in [\alpha, 1-\alpha]$.

The empirical diagnostic in Section 4 uses $p = 1/2$. Each row is divided by the arithmetic mean of its four terminal costs before averaging over supported contrasts. This is the loss of continuation-blind ordering under an equal-probability mixture, not a measurement of cumulative search regret. The $\alpha = 0.25$ sensitivity allows either continuation's mixture probability to range from 0.25 to 0.75.

B.2 CFRC safety proof details

The induction in Proposition 2 requires a finite construction horizon, deterministic adapted policies, feasible transitions, exact terminal evaluation, and an always-admissible scaffold action. State serialization includes the route prefix and remaining capacity; the remaining deviation budget is a separate memoization key. Deployment follows minimizing branches using the same deterministic policies and tie-breaking rule as evaluation.

B.3 First-divergence pruning bound

Let J be the exact optimum with all donors and (s_t, a_t) a minimizing trajectory. For a retained donor subset, let R_t contain the scaffold action and the retained proposals at s_t, subject to the remaining deviation budget. Let $Q_t(a)$ be the exact terminal cost after taking a and continuing with the full donor set and the correctly updated budget. If every $a_t \in R_t$, the optimal path survives and the restricted optimum is exactly J. Otherwise let τ be the first index with $a_\tau \notin R_\tau$. Then

$$\min_{\substack{t \le \tau \\ a \in R_t \setminus \{a_t\}}} Q_t(a) \;\le\; J_{\text{pruned}} \;\le\; J_{\text{scaffold}}. \tag{4}$$

Every retained-set trajectory first departs from the saved path at some $t \leq \tau$ via a retained alternative. Its remaining feasible continuations are a subset of those minimized by $Q_t(a)$, proving the lower bound; the scaffold supplies the upper bound. A lower bound strictly above J therefore certifies loss under joint donor removal. Tied alternatives can leave the bound inconclusive. If no donor remains, the restricted policy is exactly the scaffold. This certificate assumes the same deterministic adapted policies, state transitions and exact continuation values as Proposition 2.

C Negative Controls and Scope

Complete-program and anchor controls. LLM-only fixed-base composition is sensitive to base choice and does not reliably beat GP-320. Multiple anchors improve robustness, but matched GP anchors absorb most of the gain. Reference-free program switching also loses to GP, supporting the scaffold/residual separation.

A separate 2,880-evaluation configuration compares CFRC with GP9. GP9 deploys the eight terminal GP-320 programs and one independently selected GP-40 program as residual proposers; CFRC uses the shared selected GP-320 scaffold with eight LLM-40 donors. Both use exact $K = 2$ and account for $8 \times 320 + 8 \times 40 = 2{,}880$ successful generation evaluations. Median deployment time is $3.36\times$ GP9. API retries, extraction, selection compute, and inference time differ.

Table 17 separates these historical configurations from the two 960-evaluation confirmations in the main text. The four-pool 640-evaluation replication uses fixed donor order without the development selector.

Configuration	Generation evals.	Effect ↑	95% CI	Pool wins	Panel wins
Initial CFRC / GP9	2,880	+0.681%	[+0.275, +1.123]	1/1	5/5
CFRC / GP9-640	640	+0.717%	[+0.444, +1.017]	4/4	20/20

Table 17: **Earlier CVRP configurations.** Budgets are per arm; the generation and selection contracts differ between rows (Table 9). Endpoints are disjoint and the studies are not pooled. Positive paired log effects favor mixed proposals.

Allocation intervention. The deployment-aligned family allocator measures leave-tranche-out residual yield on a disjoint panel, then chooses a balanced or GP-heavy candidate branch. Its later chosen-versus-counter experiment uses two fresh 24-instance panels per pool. CVRP is $+0.07123\%$ with 2/4 positive pools and 5/8 positive panels. Its gate fails; the reserved confirmation remains unopened. Replay, feasibility and scaffold safety pass under equal 960-evaluation accounting, with zero new API calls.

Branch-cap intervention. A frozen confirmation keeps the scaffold, eight donors, $K = 2$ and equal 960-generation-evaluation accounting, but permits at most one ranked non-scaffold action per state in both systems. Its 480 fresh CVRP pairs across 20 panels pass replay and feasibility checks, with zero new API calls. It yields $+0.545\%$ with a $[0.017, 1.111]\%$ hierarchical interval, 3/4 positive pools, and 16/20 positive panels. The paired median mixed/GP-only suffix-evaluation ratio is $1.012\times$; the wall-time ratio remains $1.33\times$. The CVRP-specific gate passes, but the original broader joint gate did not. Unequal maximum branching alone therefore does not explain this effect.

One-alternative scale transfer. The frozen CVRP-50/100 follow-up keeps the four scaffolds, ordered donors, $K = 2$ and 960 generation evaluations per arm. Each state permits only the first non-equivalent donor action. Each size has three fresh 12-instance panels per pool (144 pairs). All 24 panels pass independently audited exactness, feasibility, safety, deviation-cap, branch-cap and zero-API checks. Table 18 reports effects and costs. Both scale gates and the joint gate fail because intervals cross zero. Paired median mixed/GP-only suffix-evaluation ratios are $1.011\times$ and $1.007\times$, while paired median wall-time ratios are $1.75\times$ and $1.85\times$ at 50 and 100 customers.

CVRP	Mixed cost	GP-only cost	Effect	Hierarchical 95% CI	Pools/panels	Suffix evals.	Time (s)
50	1556.75	1562.14	+0.352%	$[-0.057, 0.802]$	4/4; 9/12	2318.5/2253	11.24/6.52
100	2861.36	2868.74	+0.261%	$[-0.025, 0.621]$	4/4; 10/12	9515/9353.5	205.94/101.15

Table 18: **Frozen one-alternative CVRP scale transfer.** Cost columns are pooled means; suffix evaluations and times are per-arm medians in mixed/GP-only order. Positive effect is the mean paired $100 \log(\mathrm{cost}_{\mathrm{GP}}/\mathrm{cost}_{\mathrm{mixed}})$; intervals resample pools, panels, and instances. Neither row passes its predeclared positive-lower-bound gate.

Equal-deadline operating point. A separately frozen, zero-API CVRP-50 comparison evaluates the same GP scaffold with mixed8, GP16, and basis8 proposal sets under exact $K = 1$. Four pools each contribute three fresh 12-instance panels (144 paired instances). Each arm has a 30-second hard deadline: scaffold computation is charged to every arm, exact search returns its verified solution only if it finishes before the deadline, and a timeout returns the scaffold solution. Compilation is offline. The shared cap does not imply equal consumed time. The prespecified hierarchical analysis resamples pools, panels, and instances (20,000 draws). Mixed8 gains $+0.203\%$ over GP16 (95% CI $[-0.460, 0.884]$; 3/4 positive pools, 8/12 panels) and -0.300% over basis8 ($[-0.845, 0.154]$; 2/4 pools, 2/12 panels). Both quality gates fail. Mixed8 times out on 4/144 calls; neither control times out. Its median paired consumed-time ratios are $1.25\times$ versus GP16 and $1.28\times$ versus basis8. Feasibility, exact evaluated/deployed equality, scaffold safety, and the deviation cap pass on every returned route.

Other domains. JSSP has 106/641 reversing deviations; 24/70 baseline-negative deviations inside improving endpoints become beneficial. BinPack programs choose the same bin or bins with equal remaining capacity, leaving no non-equivalent action support to identify reversals. Algorithm-level CFRC evidence is CVRP.

Peer deployment contracts. Native EoH, MCTS-AHD, and HiFo deploy one training-selected program; CFRC deploys nine with exact continuation search. At a separate 640-evaluation CVRP operating point, CFRC gains 7.505% against the primary native peer on five positive fresh panels, at different deployment capacity and cost. EoH-S executes five heuristics per instance and keeps the best; deploying one member changes its contract, so it is excluded from the main ranking.