What a Permutation Representation Is Allowed to Learn: Objective Structure, Probabilistic Model Design, and Selection Regime from Scalar to Tri-Objective Permutation Flow-Shop Scheduling

Anonymous Author(s)*

Abstract

An optimization algorithm does not learn a scheduling problem directly; it learns only the structures its representation is capable of encoding. This paper tests that claim for permutation flow shop scheduling across three selection regimes of increasing objective complexity: scalar selection on a single objective, bi-objective Pareto selection on makespan and total flow time, and a tri-objective extension that adds non-redundant internal machine idle time as a third, structurally distinct objective. We compare twelve probabilistic permutation models — eight incremental COIN-family estimation-of-distribution algorithms and four rebuilt edge/node-histogram EDAs, with and without punched-template inheritance — under matched population and generation budgets. Under Pareto selection (Taillard TA001–TA030, thirty seeds, 10,800 completed runs, no recorded failures), SNE-COIN attains the best mean normalized hypervolume (0.9871 at reference point $(1.1, 1.1)$); NB-COIN attains the best mean IGD+ (0.1096) against a pooled observed reference front; five of six incremental representations outperform the strongest rebuilt-histogram baseline in mean hypervolume. A Friedman test rejects equal performance across all twelve methods ($\chi^2 = 8040.47$, 900 instance-seed blocks, $p < 10^{-300}$), and the leading method remains significantly ahead of its closest competitor after Holm correction (median $\Delta\text{HV} = 0.0186$, $p = 0.000183$). Punched-template inheritance improves the rebuilt edge histogram but degrades the rebuilt node histogram and both incremental

*Affiliation withheld for review. **[VERIFY: institution, department, contact details before submission.]**

COIN hosts it is grafted onto — a negative result that rules out "templates help estimation-of-distribution algorithms" as a general claim. Under scalar selection (TA001–TA020, four seeds, 1,920 completed runs, 400 generations, equal evaluation budgets), the same representational advances do *not* uniformly carry over: punched-template methods (NHBSA/WT, NB-COIN/WT, EHBSA/WT) rank at or near the top for both makespan and total flow time, while SNE-COIN, HNE-COIN, and CNE-COIN — the representations that lead under Pareto selection — occupy the middle of the scalar ranking, and COIN and EHBSA/WO rank last on both objectives. We extend this two-regime interaction with an exploratory tri-objective study (makespan, total flow time, and internal machine idle time jointly), currently at three confirmed seeds and a partially completed eight-of-ten-seed exploratory batch — evidentiary standards well short of this paper's thirty-seed bi-objective confirmatory benchmark, and explicitly reported as such. The tri-objective evidence is presented only as a motivating pattern consistent with representational complementarity across objectives, pending a twenty-seed confirmatory run that has not yet been executed. We interpret the confirmed two-regime contrast, and the exploratory three-regime extension, through a common frame: objective structure, learned representation, and selection regime interact, rather than any one representation being unconditionally best, and adding a third non-redundant objective is a stress test of that interaction rather than a settled result in its own right. The contribution is controlled evidence about probabilistic model design across selection regimes of increasing objective complexity, not a claim to match the strongest existing multi-objective or single-objective permutation flow-shop scheduling solvers.

Keywords: multi-objective permutation flow shop scheduling, estimation of distribution algorithm, COIN algorithm, hypervolume, IGD+, makespan, total flow time, internal machine idle time, Taillard benchmark, Pareto optimization, scalar selection, tri-objective optimization, probabilistic model

---

1. Introduction

An algorithm does not fail only because it learns too little. Sometimes it learns the wrong kind of regularity for the question being asked. This is easy to overlook in permutation flow-shop scheduling (PFSP), where algorithms are usually judged by a single final number. For makespan or total flow

time alone, the winner is simply the method that reaches the lowest value. Under two objectives, however, "good" no longer describes one schedule. It describes a set of schedules that preserve different compromises, and that change in the meaning of good may also change which probabilistic representation is useful for multi-objective permutation flow shop scheduling.

A permutation search cannot rely, unmodified, on the schema theory Holland developed for fixed-locus binary strings. When Goldberg and Lingle introduced partially matched crossover for the traveling salesman problem, they were responding to exactly this mismatch: a fixed-locus schema is defined by which allele occupies which locus, but a permutation's useful building blocks are not tied to a locus in that sense — they are order relations among jobs, adjacency relations between consecutive jobs, or the identity of the job occupying a particular absolute position [34]. Later survey work made this vocabulary explicit without collapsing it back into the binary case: Larrañaga et al.'s review of TSP genetic operators distinguishes an *adjacency* representation, in which the building block of interest is which job follows which, from *ordinal* and *position-based* operators, in which the building block is a job's rank or its absolute slot [35]. A permutation representation is therefore a choice about which of these events — adjacency, order, or position — a model is allowed to count, and a model can also mix them; a model that only counts adjacency events cannot represent an absolute-position regularity without additional machinery, and the reverse holds for a pure position model. This is the representational vocabulary the rest of this paper uses, and it is the reason "EDA" is too coarse a label for what follows: two permutation EDAs that share this vocabulary can still differ sharply in what they count.

COIN treats this representational choice as a learning problem rather than a fixed crossover-operator design. Instead of encoding one representation into an operator, COIN estimates a probability model over whichever event class it is built to count — directed adjacency, in its original edge-based form — from a population of candidate permutations, and updates that model with *signed* evidence: structures observed in a high-quality cohort are rewarded and structures observed in a low-quality cohort are explicitly punished, with the update persisting and decaying across generations rather than being rebuilt from scratch every generation [29]. This is the conceptual bridge from permutation schema theory to the COIN probability model: where classical schema theory asks which building blocks survive crossover, COIN asks which building blocks are worth remembering, for how long, given both what

has worked and what has not.

Before describing the representational variants this paper compares, it is necessary to separate two things that are easy to conflate: what a model counts, and how it learns and samples what it counts. Two matched baseline pairs make the distinction concrete. COIN and EHBSA both represent directed adjacency evidence in a matrix indexed by (previous job, next job), and both construct a complete, repair-free permutation by roulette-sampling an unused successor at each step; despite the shared representation, they are not the same learning process. COIN is an incremental, persistent learner: a small rewarded cohort (5% of the population) reinforces the edges it contains, an equally small punished cohort suppresses the edges it contains, the resulting integer weights carry across generations, and every legal transition retains some exploration weight rather than being driven to zero. EHBSA is a generational empirical estimator: every generation it discards its previous model and rebuilds a smoothed edge histogram from the top 50% of the current population, so what it "remembers" is entirely determined by the current elite cohort and nothing earlier [12]. EHBSA can additionally be run with a punched parent template (EHBSA/WT: a selected parent has 50% of its positions marked for resampling from the histogram, with one-to-one replacement) or without one (EHBSA/WO: a complete permutation sampled fresh from the histogram). NB-COIN and NHBSA form the analogous pair for position–job evidence, indexed by (position, job) rather than (previous job, next job): NB-COIN again retains and incrementally updates persistent weights from its reward/punish cohorts, while NHBSA rebuilds a smoothed position histogram from the elite cohort every generation, with the same WO/WT sampling distinction [14]. Table 1 makes the resulting three-way distinction explicit: *representation* asks which event is counted (edge or position); *learning dynamics* asks how evidence is retained and updated (persistent signed memory versus a histogram rebuilt from scratch); and *sampling/replacement* asks whether a real parent template survives into the offspring. A performance difference between, say, COIN and EHBSA cannot be attributed to representation alone, because their learning dynamics and (for the WT variants) their replacement mechanism also differ — a point this paper returns to directly when the direct COIN/WT and NB-COIN/WT ablations are reported (Section 5).

Only after this factorization does it make sense to ask what a new variant adds. Each of the six COIN-family representations compared in this paper answers a specific question about what the two baseline event classes,

Table 1: The two matched baseline pairs: sharing a representation does not mean sharing a learning process. WO/WT sampling is a further axis, present identically in both members of the histogram pair and grafted onto the COIN pair as a direct ablation (Section 3.3).

Method	**Event counted**	**Model persistence**	**Positive & negative evidence**	**Sampling / replacement**
COIN	directed edge (prev. job → next job)	incremental, signed, decays across generations	explicit reward (top 5%) *and* punish (bottom 5%)	no native template; WT is a direct ablation
EHBSA	directed edge	rebuilt from top 50% every generation	positive (elite frequency) only	WO: none; WT: 50% punched parent, one-to-one
NB-COIN	position → job	incremental, signed, decays across generations	explicit reward *and* punish	no native template; WT is a direct ablation
NHBSA	position → job	rebuilt from top 50% every generation	positive (elite frequency) only	WO: none; WT: 50% punched parent, one-to-one

alone, cannot express. Plain COIN encodes directed adjacency/coincidence evidence but not which job may legitimately open the sequence. NB-COIN encodes position–item evidence instead of adjacency, trading one blind spot for another. CNB-COIN keeps NB-COIN's position model but fills positions in a fixed left-to-right construction order rather than a randomized one, isolating construction order from the probability table itself. SNE-COIN keeps the edge model but adds a separately learned starting-node distribution, so the first-job choice is no longer left to chance. HNE-COIN and CNE-COIN both combine node and edge evidence, but by different mechanisms: HNE-COIN generates a node-model scaffold with scattered anchors and completes the remaining holes from edge probabilities, while CNE-COIN makes an independent node-or-edge choice at every position, link by link, with no scaffold

or punched parent involved. Section 3 gives each of these representations precisely enough to be re-implemented; this paragraph states only what information each one adds relative to the pair it is compared against.

This representational lineage did not begin as, and was not converted from, a single-objective method. Its first published formulation was already multi-objective: the author identifies the 2009 IEEE Congress on Evolutionary Computation paper introducing COIN as the first COIN paper, not a later multi-objective extension of an earlier scalar method [29]; that paper's multi-objective mechanism adapts non-dominated sorting so that "not-good" solutions are redefined relative to the opposite side of the objective space, rather than combining objectives into one scalar score. We have not located an earlier Wattanapornprom/Chongstitvatana publication describing COIN, which is consistent with, though it does not by itself prove, that designation. The signed reward/punish learning rule itself was then formalized in the author's dissertation and its journal treatment [30, 31], an edge-based variant was evaluated on multimodal combinatorial puzzles [32], and a node-based variant was applied to Sudoku, which the present paper treats as the earliest node-based COIN application we have located, subject to the same caveat that we searched but cannot exhaustively rule out an earlier one [33]. Sudoku motivates position/node evidence specifically because its building blocks — a digit's validity within a row, column, and box — depend on *where* a value sits, not on which value precedes it, which is exactly the representational gap plain edge evidence cannot close. A related line of work on hybrid negative-correlation learning for combinatorial optimization is part of the same diversity/negative-learning lineage [37]; we were not able to access that paper's full text in this revision and therefore cite it only for its title-level connection to signed, negative-evidence learning, not for any specific technical mechanism. Node-based COIN reached permutation flow-shop scheduling specifically through Srimongkolkul and Chongstitvatana, who applied it to total-flow-time minimization [24] and, in a companion study, to makespan minimization [25], both under a single scalar objective — the point at which, historically, this representational family narrowed from the original multi-objective setting to a scalar one. A separately authored line of work later applied node-based estimation-of-distribution ideas to an order-acceptance capacity-balancing problem outside PFSP [36]; that paper's first author is Watcharee Wattanapornprom, a different researcher from the present manuscript's author (Warin Wattanapornprom, third of four authors on that paper), and we were unable to access its full text in this revision, so

we cite it only as part of the extended publication record around this representational family and do not attribute a specific starting-node mechanism to it without having read the paper directly.

The present paper's multi-objective setting therefore does not depart from COIN's origin so much as return to it, after the representational family summarized above had matured: COIN began as a multi-objective method, moved through a line of scalar, single-objective work including its PFSP application, and is evaluated here again under genuine Pareto selection — but now with a systematic representational comparison (edge, node, start-node, and two hybrid mechanisms, with and without directly grafted template inheritance) that neither the original multi-objective introduction nor the intervening scalar work provided. Stated as a hypothesis rather than a settled result, the reason to expect multi-objective selection to matter here is not simply that "more objectives are harder." It is that multiple objectives can expose complementary structural evidence to a learned representation and help it retain structurally different, mutually useful solutions, rather than collapsing every generation toward one scalar winner. One piece of existing evidence makes this plausible without proving it: the same signed, negative-correlation learning rule used by every incremental COIN variant in this paper was originally motivated by multimodal puzzle solving, where negative-sample learning was reported to help prevent premature convergence, promote diversity, and preserve good building blocks across multiple useful modes rather than collapsing to one [31]. A mechanism designed to preserve multiple useful modes under multimodal *single*-objective selection is a reasonable candidate mechanism for also preserving multiple useful trade-offs under *multi-objective* selection — but multimodal optimization and multi-objective optimization are different problems, and this paper treats the connection as a motivating hypothesis for why COIN's representations might behave well under Pareto selection, not as a proof that the two phenomena are the same. The single-objective experiment reported in Section 5.5 is a direct empirical check on part of this picture: it asks whether the representational advances that help under Pareto selection also help under scalar selection of either objective alone, and, as summarized below, the answer is no, not uniformly — which is itself informative about what multi-objective selection is and is not doing for these representations.

Makespan and total flow time provide a direct way to test the Pareto-selection side of this picture. Both are standard PFSP objectives, but they reward different job orders: makespan emphasizes completion of the entire

production schedule, whereas total flow time emphasizes earlier completion across individual jobs. Reducing them jointly under Pareto optimization requires an algorithm to retain alternatives along a Pareto front instead of repeatedly concentrating probability around one scalar winner. A model that is mediocre at collapsing onto a single optimum may still be good at preserving useful, competing structures; conversely, a model that converges efficiently under scalar selection may discard diversity that becomes valuable under Pareto selection. Neither outcome follows automatically from the definition of an EDA or a probabilistic model over permutations; it has to be measured, and, together with the matched scalar experiment, it is the central object of study in this paper. We state the working hypothesis as: *an optimization algorithm does not learn the scheduling problem directly; it learns only the structures that its representation is capable of seeing*, and which structures matter depends on how the objective is defined and how candidates are selected. The short version, if a slogan is wanted, is that representation is not a detail an EDA hides from the reader — it is the experiment.

We therefore compare twelve probabilistic permutation models under two selection regimes and matched evaluation budgets: Pareto selection over makespan and total flow time jointly, and scalar selection of each objective separately. The comparison includes incremental edge, node, consecutive-node, start-node edge, and two node–edge hybrid COIN models; direct punched-template versions of Edge COIN and NB-COIN; and EHBSA and NHBSA with and without templates. The algorithms share feasible-permutation sampling, population size, generation budget, and (within each regime) instances and seeds; their representational and learning mechanisms remain deliberately visible rather than being hidden behind one generic "EDA" label.

Under Pareto selection, the experiment comprises 30 independent seeds on Taillard instances TA001–TA030, with population 100 and 400 generations for every method and instance, totaling 10,800 completed multi-objective runs with no recorded run failures. SNE-COIN obtains the highest mean normalized hypervolume, while NB-COIN obtains the lowest (best) IGD+. Five of the six incremental COIN representations exceed the strongest classical histogram baseline in mean hypervolume. At the same time, the experiment rejects a convenient universal explanation: punched templates improve EHBSA but degrade NHBSA and both incremental models to which the mechanism is directly attached. Under scalar selection, the experiment comprises 4 independent seeds on Taillard instances TA001–TA020, with the same population, generation budget, and per-run evaluation count, totaling

1,920 completed single-objective runs (960 per objective) with no recorded run failures; here, punched-template methods — not the newer COIN representations — occupy the top ranks for both makespan and total flow time, and COIN and EHBSA/WO rank last on both. The useful distinction is therefore not simply "template" versus "no template," nor "COIN" versus "histogram," nor even "multi-objective versus single-objective is always better for representation X." It is the interaction between what a model represents, how evidence persists across generations, and how the selection regime — Pareto or scalar — defines a promising population.

This paper makes five contributions. First, a controlled, equal-budget, 30-seed comparison of twelve incremental and rebuilt-histogram EDAs for jointly minimizing makespan and total flow time under Pareto optimization. Second, a matched, equal-budget, four-seed comparison of the same twelve representations under scalar selection of makespan and of total flow time separately, directly testing whether the Pareto-selection ranking generalizes. Third, a representational factorization — edge, position, start-position, and two distinct hybrid mechanisms, plus random-versus- consecutive position-fill order and rebuilt-versus-incremental learning dynamics — that isolates what a model learns from how it learns it, so that representational choices, rather than tuning differences, explain the observed performance gaps in both regimes. Fourth, a reproducible statistical analysis reporting normalized hypervolume, IGD+, paired ranks, runtime, and Holm-corrected significance tests over 900 multi-objective instance-seed blocks, together with paired rank and ARPD statistics over 80 single-objective instance-seed blocks per objective, using a pooled observed reference front for the multi-objective case. Fifth, a non-redundant tri-objective extension — adding internal machine idle time to makespan and total flow time — reported at its true evidentiary stage (an exploratory three-seed pilot and a partially completed eight-of-ten-seed batch, both short of a confirmatory sample), rather than inflated to match the confidence of the first four contributions. We pose three research questions that this paper now answers directly:

- **RQ1.** Holding search engine, budget, and selection rule fixed, does the class of structure a permutation representation can encode (job adjacency, position assignment, or explicit start position) determine Pareto-approximation quality under joint makespan/total-flow-time selection?
- **RQ2.** Does punched-template inheritance transfer its benefit across

representation families (rebuilt histogram versus incremental signed learning), or is its effect specific to the family it was designed for?

- **RQ3.** Does the representation ranking observed under Pareto selection match the ranking observed under scalar selection of makespan or total flow time alone, or does the selection regime itself change which representation performs best?

Section 5.5 reports the RQ3 comparison directly. We are explicit about its evidentiary weight relative to the 30-seed multi-objective result: four seeds is the current confirmatory single-objective experiment, not a definitive journal-level statistical claim on its own, and we report it, and label it, accordingly (Section 7).

Makespan and total flow time are not the only structurally meaningful objectives in permutation flow-shop scheduling. Internal machine idle time — the idle time a schedule inserts *between* a machine's first and last processed job, as opposed to the idle time before the first job starts or after the last job finishes — is a third quantity a scheduler can explicitly minimize, and it is not a disguised restatement of makespan: only the machine's *total* horizon idle time is an affine function of makespan and therefore redundant with it (Section 4 gives the precise definitions and this distinction in full); internal idle time is not affine in makespan, because it depends on where processing gaps fall within the schedule, not only on the schedule's overall span. A tri-objective extension — makespan, total flow time, and internal machine idle time jointly — is therefore a genuine stress test of this paper's central interaction claim, not a cosmetic addition of a third axis: if objective structure, representation, and selection regime interact as this paper argues for two objectives, adding a third, non-redundant objective should make that interaction more visible, not less, because a representation that already struggles to retain heterogeneous evidence under two competing objectives has a harder task under three. We pose two further research questions for this extension, and answer them only provisionally:

- **RQ4.** Does adding a third, non-redundant objective (internal machine idle time) to the existing bi-objective makespan/total-flow-time selection alter or amplify the representation-dependent performance already observed under two objectives?

- **RQ5.** Do different learned representations contribute complementary regions of the observed three-objective Pareto front, consistent with

representation-specific structural coverage rather than one representation dominating all three objectives simultaneously?

We answer RQ4–RQ5 only provisionally in this paper (Section 5.6, Section 6.2): the tri-objective evidence base currently available — three confirmed seeds, plus a separate, partially completed eight-of-ten-seed exploratory batch — is well short of the thirty-seed standard this paper's own bi-objective result meets, and we report the tri-objective pattern as motivating rather than confirmatory, pending a twenty-seed run at the same protocol.

The remainder of this paper is organized as follows. Section 2 positions the study against multi-objective PFSP and permutation-EDA literature and states the research gap. Section 3 defines the twelve compared representations precisely enough to be re-implemented. Section 4 specifies the experimental protocol, metrics, and statistical procedure, including the tri-objective extension's protocol and evidentiary status. Section 5 reports the multi-objective confirmatory results, Section 5.5 within it reports the matched single-objective comparison, and Section 5.6 reports the exploratory tri-objective extension. Section 6 interprets the confirmed regimes together, distinguishing what is measured from what is hypothesized, and Section 6.2 does the same for the tri-objective extension specifically. Section 7 states limitations and threats to validity, Section 8 states the concrete next steps, and Section 9 concludes.

2. Related work and research gap

2.1. Scope and positioning

Permutation flow-shop scheduling (PFSP) is a canonical sequencing problem in which every job visits the machines in the same order and a single permutation determines the processing order on all machines. Makespan minimization ($C_{\max}$) represents production-horizon efficiency, whereas total flow time (TFT, or total completion time when release dates are zero) captures the time accumulated by all jobs in the system. The objectives are related but not interchangeable: a sequence that finishes the last job early can still delay many intermediate completions, while one that reduces accumulated completion time need not minimize the final completion time. Treating $C_{\max}$ and TFT jointly therefore changes what information a search must preserve; it is not merely a larger version of either scalar problem.

Taillard's benchmark suite established the experimental backbone for much of the PFSP literature [1]. It remains useful because the same instance families have supported single- and multi-objective studies. Familiarity alone, however, does not guarantee a fair comparison: population size, stopping budget, independent runs, reference-front construction, and indicator normalization all affect the outcome. The present comparison accordingly fixes population and generation budgets across twelve EDAs and uses repeated runs on the same thirty instances.

2.2. From scalar flow-shop search to Pareto search

Early multi-criteria PFSP methods often transformed several criteria into one weighted or normalized objective. Aggregation is convenient but requires preferences before the search and can miss unsupported parts of a non-convex trade-off front. Later work moved toward a posteriori methods that return a set of non-dominated schedules. Varadharajan and Rajendran developed multi-objective simulated annealing specifically for makespan and total flow time [2]. Framinan and Leisten proposed multi-objective iterated greedy search that constructs and improves a working set of Pareto-efficient schedules [3]. Minella, Ruiz, and Ciavotta evaluated twenty-three algorithms over several objective pairs [4] and later introduced restarted iterated Pareto greedy (RIPG), strengthening this line through restart and Pareto-set management [5].

Other influential studies followed evolutionary and memetic routes. Pasupathy et al. applied a multi-objective genetic algorithm to makespan and total flow time [6], and Yagmahan and Yenisey used ant-colony optimization [7]. Lin and Ying proposed a bi-objective multi-start simulated-annealing method [8]. Chiang, Cheng, and Fu's NNMA combined NSGA-II with an NEH-derived local improvement procedure and reported strong results over ninety Taillard instances [9]. Collectively, these studies show that the $C_{\max}$–TFT formulation is established and that strong performance has commonly come from combining Pareto selection with problem-specific construction or local search. We name these methods explicitly because they define the performance ceiling this paper does not attempt to test (Section 7); the present study is a controlled representation comparison among EDAs at fixed budget, not a claim to match RIPG or NNMA.

Two reviews delimit the field. Minella et al. supplied both a review and a controlled evaluation rather than relying only on values copied from unrelated experiments [4]. Yenisey and Yagmahan later classified eighty-six

multi-objective flow-shop studies and observed a shift toward contemporary metaheuristics and richer formulations [10]. They also expose an important distinction for this study. The mainstream question is which complete metaheuristic framework produces the strongest approximation set. Our narrower question is: when the search engine and computational budget are held constant, which permutation statistics should a probabilistic model learn? This is a representation study, not a claim to replace the strongest local-search-based PFSP solvers.

2.3. Estimation of distribution in permutation spaces

Estimation-of-distribution algorithms (EDAs), also called probabilistic model-building genetic algorithms, replace conventional crossover and mutation with explicit distribution estimation and sampling. In permutation domains, ordinary independent-variable models do not automatically preserve the all-different constraint. A useful model must describe regularities in promising permutations while permitting efficient sampling of valid offspring.

Tsutsui introduced edge-histogram modeling for sequence problems, representing how often one item follows another [11]. Tsutsui and Miki then applied it to flow-shop scheduling and distinguished sampling without a template (EHBSA/WO) from sampling with a partially preserved parent template (EHBSA/WT) [12]. The template acts as structured inheritance: part of a selected permutation remains fixed while missing jobs are sampled from the model. Later studies examined the tag-node idea and compared edge with node histograms [13, 14]. A node histogram estimates position–job associations rather than adjacencies; EHBSA and NHBSA consequently retain different information even when their evolutionary wrappers resemble one another. We do not claim any implementation equivalence between the tag-node mechanism and the start-node representation studied here (Section 3); the resemblance is at the level of "both encode something about the starting position," not at the level of a validated shared mechanism.

The distinction matters in PFSP. Edge models express neighboring-job building blocks but do not explicitly represent the first job unless a start marker or equivalent device is included. Node models express absolute position but not direct adjacency. A template can preserve higher-order structure that neither marginal model expresses, but it can also restrict exploration. These are representational trade-offs, not a universal hierarchy.

PFSP EDA research continued beyond the original histograms. Jarboui et al. combined distribution estimation with variable-neighborhood search for total-flow-time minimization [15]. A later position-oriented EDA used permutation inheritance and local improvement for the same objective [16]. Random-key and hybrid EDAs have also appeared. More recently, PGS-EDA (Position-Guided Sampling EDA) was compared with random-key EDA, UMDA, GM-EDA (Generalized Mallows EDA), NHBSA, and EHBSA on Taillard makespan instances [17]. This close modern comparator confirms that probabilistic representation remains an active issue. Yet these studies are predominantly scalar, and strong variants often include local search. They do not isolate how edge, node, start-node, and hybrid statistics behave when selection uses Pareto dominance over $C_{\max}$ and TFT.

A separate line of permutation EDAs represents a distribution through explicit ranking or ordering statistics rather than through pairwise- or positional-frequency counts, and is worth distinguishing here from the edge/node framework this paper builds on, since none of it is included in our own twelve-method comparison. The Generalized Mallows model represents a permutation distribution as a distance-based ranking centered on a consensus permutation, with per-position dispersion parameters controlling how tightly each rank is held around that consensus; Ceberio et al. introduced this model for combinatorial EDAs [18] and later applied it directly to PFSP, reporting competitive results against established permutation EDAs [19]. Because a single Mallows model is unimodal around one consensus permutation, Ceberio et al. later proposed a kernel/mixture construction that averages several Mallows models, each centered on a different reference permutation, which breaks the single-mode assumption [20]; this directly targets the structural-averaging limitation we raise for pairwise/positional models in Section 6.1, though we were not able to confirm from accessible metadata whether that paper's own experiments cover PFSP specifically, and we do not assume so. The Plackett-Luce model instead represents a permutation as a sequence of choices, with each remaining job selected with probability proportional to a learned "worth" score — a middle ground between an explicit position model and a pure ordering model; Ceberio et al. also introduced this ranking model for permutation-based optimization [21]. Random-key EDAs take a different route again: rather than modeling the permutation directly, they model a continuous latent priority value per job (e.g., via a multivariate Gaussian) and decode a permutation by sorting jobs on the sampled priorities, borrowing continuous-EDA machinery; Ayodele et al. introduced

this combination [22] and later applied it to PFSP total-flow-time minimization specifically [23]. None of these four families — Generalized Mallows, Mallows-kernel/mixture, Plackett-Luce, or random-key — is part of this paper's own comparison: they occupy a genuinely different design axis (global ranking/ordering statistics, or a continuous latent encoding, rather than pairwise edge or positional node counts) and would need their own controlled, equal-budget comparison against the COIN and histogram families studied here, under Pareto selection specifically, before any claim could be made about how they interact with it. We state this as a scope boundary rather than an oversight and return to it in Section 8. (We also searched for a longest-common-subsequence-based EDA that would sit alongside these as a template-like, elite-string-inheritance mechanism; we did not find a credible, sufficiently-cited primary source for such a method under any name we searched, and we do not cite one here rather than risk citing a paper that does not establish what its name would imply.)

2.4. COIN as incremental signed evidence

Coincidence algorithms (COIN) occupy a related but distinct point in this design space, and their lineage is worth stating precisely because it is easy to compress into "COIN is a flow-shop method," which understates both its origin and its scope. COIN was introduced by Wattanapornprom et al. as a general-purpose multi-objective combinatorial optimizer, in an edge-based form, evaluated on multi-objective combinatorial benchmarks rather than PFSP specifically [29]. The underlying learning rule — incremental, signed reinforcement of structures observed in a high-quality cohort and penalization of structures observed in a low-quality cohort, persisting and decaying across generations rather than being rebuilt from scratch — was developed as "negative correlation learning" and formalized in Wattanapornprom's dissertation and its journal treatment [30, 31]; this is the same reward/punish cohort mechanism used identically by every incremental COIN variant in Section 3. An edge-based variant was further evaluated on multimodal combinatorial puzzles [32], and a node-based variant was later applied to Sudoku [33], establishing that the representational choice (edge versus node/position evidence) was already recognized as consequential outside PFSP before this paper's comparison. Node-Based COIN reached permutation flow-shop scheduling specifically through Srimongkolkul and Chongstitvatana, who applied it to total-flow-time minimization [24] and, in a companion study, to makespan minimization [25], both under a single scalar objective. The present pa-

per's multi-objective setting therefore does not depart from COIN's origin so much as return to it: COIN began as a multi-objective method, moved through a line of scalar, single-objective PFSP work, and is evaluated here again under genuine Pareto selection — but now with a systematic representational comparison (edge, node, start-node, and hybrid evidence, with and without template inheritance) that neither the original multi-objective introduction nor the intervening scalar PFSP work provided. Like a node-histogram EDA, NB-COIN associates jobs with positions. Its model is not simply a histogram rebuilt from the selected population: COIN maintains incremental signed evidence, reinforcing relations observed in favorable solutions and penalizing those observed in unfavorable solutions. A rebuilt histogram summarizes the current selected set, whereas incremental signed learning carries path-dependent memory of earlier rewards and punishments.

This difference is hidden if algorithms are grouped only by matrix shape. An edge-frequency matrix and an Edge-COIN matrix may have the same dimensions without having the same dynamics. Transferring EHBSA/WT's punched-template mechanism to COIN likewise changes more than sampling notation: it couples persistent signed evidence with direct parental inheritance. WT's success in EHBSA cannot establish that this coupling helps COIN; the transfer requires an ablation, which we report in Section 5.

The present study evaluates eight incremental COIN variants and four rebuilt-histogram baselines under one bi-objective selection rule. The variants separate edge evidence, position evidence, consecutive position filling, explicit start-node information, two distinct edge–node hybrid mechanisms, and direct template transfer. This factorial interpretation is more informative than a broad contest between names: it asks which relations are learned, how offspring are completed, and whether a template complements or conflicts with model memory.

2.5. Evaluation of approximation sets

A multi-objective result cannot be summarized by the best value of either objective alone. Hypervolume (HV) jointly rewards convergence and spread relative to a reference point and is Pareto-compliant under the usual conditions [26]. IGD+ measures distance from a reference set to the approximation while modifying ordinary IGD to obtain weak Pareto compliance [27]. Reporting both matters because one method can cover more dominated volume while another remains closer to the empirical reference front. Both indicators still depend on transparent normalization and reference-front construction,

and both are computed here against a front pooled from the very methods being compared — a dependency we return to explicitly in Section 7 rather than treat as a neutral choice.

Across instances, ranks and paired non-parametric tests are preferable to unqualified grand means. The Friedman test addresses a global null hypothesis across algorithms, and paired Wilcoxon tests with multiplicity correction can localize differences [28], an approach originally developed for comparing classifiers across data sets and adapted here to comparing optimizers across instance-seed blocks. Here, the union reference front is constructed from all algorithms and seeds within each instance before normalized HV and IGD+ are calculated. It is an empirical, pooled, observed reference front, never asserted as the unknown true Pareto front.

2.6. Research gap and defensible contribution

The literature establishes four facts. First, makespan and total flow time are a mature PFSP objective pair. Second, sophisticated Pareto and memetic algorithms already provide strong baselines (RIPG [5], NNMA [9], multi-start SA [8], among others). Third, edge and node histograms are established permutation models, with punched-template inheritance (WT) as a model-dependent device rather than a universal one. Fourth, COIN began as a multi-objective combinatorial optimizer [29] but its representational variants have since been developed and evaluated almost entirely under scalar objectives, including its permutation flow-shop applications to total flow time [24] and to makespan [25]; to our knowledge, no prior work has re-evaluated the resulting family of COIN representations as a multi-objective PFSP method.

What remains insufficiently studied is the interaction between permutation representation and Pareto selection under a controlled EDA budget. Prior evidence does not determine whether persistent signed learning benefits from edge, position, or explicit start-position information; whether hybridizing those statistics improves front quality; or whether punched-template inheritance transfers safely from rebuilt histograms to incremental COIN. Within the literature we reviewed, we found no controlled, equal-budget study that holds search engine and selection rule fixed while varying what a permutation representation is allowed to encode, under genuine Pareto selection, across incremental and rebuilt probabilistic models and their template-inheritance variants. The present study addresses that gap. Its contribution is neither the invention of multi-objective PFSP nor an overall state-of-the-art claim against RIPG- or NNMA-class solvers. It is controlled evidence

that representation and learning memory interact with multi-objective selection — and that a mechanism beneficial to one model family can harm another. A further, narrower gap motivates this paper's tri-objective extension (Section 4.6): within the same literature we reviewed, comparisons of permutation EDA representations under three or more jointly optimized, non-redundant objectives are essentially absent, so the bi-objective interaction this paper establishes has, to our knowledge, no existing three-objective counterpart to compare against, whether confirmatory or exploratory.

Table 2 condenses the evidence matrix maintained alongside this paper (full matrix retained as supplementary material) to show, at a glance, where each cited study sits relative to the present design.

Table 2: Condensed literature-evidence matrix (full matrix in supplementary material). "Obj." abbreviates objective(s); $C_{\max}$ = makespan, TFT = total flow time.

Study	**Obj.**	**Method**	**Template / LS**	**Relevance to this paper**
Varadharajan & Rajendran [2]	$C_{\max}$, TFT	MOSA	simulated annealing	Direct objective-pair precedent
Minella et al. [5]	MOFSP	RIPG	restart + iterated greedy	Strong Pareto baseline (context, not competitor)
Chiang et al. [9]	$C_{\max}$, TFT	NNMA	NSGA-II + NEH	Strong memetic boundary (context)
Tsutsui & Miki [12]	scalar	EHBSA	WO/WT lineage	Direct histogram-PFSP origin
Tsutsui et al. [14]	scalar	EHBSA vs. NHBSA	sampling comparison	Direct representation comparison
Jarboui et al. [15]	TFT	EDA	variable-neighborhood search	Scalar hybrid EDA
Srimongkolkul & Chongstitvatana [24]	TFT	NB-COIN	incremental signed update	Direct COIN lineage (total flow time)
Srimongkolkul & Chongstitvatana [25]	$C_{\max}$	NB-COIN	incremental signed update	Direct COIN lineage (makespan)
Ceberio et al. [19]	scalar (FSP)	GM-EDA	distance-based ranking, no template	Alternative ranking-model representation, not compared here
Ceberio et al. [20]	general	Mallows-kernel EDA	mixture of Mallows models	Addresses the multimodal-averaging limitation (§6.1)
Ceberio et al. [21]	general	PL-EDA	sequential selection, no template	Ranking model between position and order
Ayodele et al. [23]	TFT	RK-EDA	continuous latent priority + sort	Continuous-encoding alternative to explicit permutation models
Lemtenneche et al. [17]	$C_{\max}$	PGS-EDA	position-guided sampling	Recent PFSP EDA baseline; overlaps somewhat with NHBSA
Ishibuchi et al. [27]	indicator	IGD+	weak Pareto compliance	Primary IGD+ definition
Demšar [28]	statistics	Friedman & Wilcoxon	post-hoc correction	Statistical method, adapted from data sets to instances

3. Algorithms and representations

All twelve compared methods generate feasible permutations by masking jobs already placed, so infeasibility is structurally impossible and every evaluated individual is a legal schedule. All twelve share the same multi-objective selection rule: candidates are ordered first by nondominated depth and then by descending crowding distance before either model family accumulates its statistics; no weighted sum of makespan and total flow time is used at any point. This section defines each representation precisely enough to be re-implemented, states the confirmatory hyperparameters, and gives the naming used in this paper alongside the working names used during development.

3.1. Shared COIN learning loop

For every generation, each incremental COIN variant follows the same outer loop, implementing the negative-correlation (reward/punish cohort) learning rule introduced for combinatorial optimization by Wattanapornprom et al. and formalized in [29, 30, 31]:

$P \leftarrow$ sample 100 feasible permutations from the current model
$F \leftarrow$ evaluate every permutation in P
score $\leftarrow$ Pareto-depth/crowding order (never a weighted sum)
$R \leftarrow$ top 5% of P by score
$U \leftarrow$ bottom 5% of P by score
reward structures observed in R; punish structures observed in U
update the model while retaining positive exploration weight
archive unique nondominated schedules (capped only by dominance)

The confirmatory settings are: reward-cohort ratio 5%, punishment-cohort ratio 5%, training (learning) rate 5, population 100, and 400 generations, applied identically to every incremental variant. The four rebuilt-histogram baselines (EHBSA, NHBSA) instead reconstruct their empirical model from the top 50% of the population every generation, with a bias ratio of 0.005; they carry no memory across generations beyond that just-rebuilt model. This is the central representational contrast the paper investigates: rebuilt-every-generation statistics versus signed, path-dependent statistics that persist and decay over time.

3.2. COIN-family representations

COIN / Edge-Based COIN [29, 32]. State: a directed matrix $W[i, j]$ (weight from job i to job j), including a wraparound entry from the last job back to the first. Sampling: choose the first job uniformly at random, then repeatedly roulette-sample an unused successor from the current job's row. Learning unit: directed job-to-job edges. This representation does not explicitly encode which job opens the permutation as a separate learned event — the first-job choice is uniform by construction, independent of the edge statistics.

NB-COIN (Node-Based COIN). State: a matrix $W[p, j]$ (weight of job j at position p). Sampling: shuffle the order in which positions are filled, then roulette-sample an unused job at each selected position. Learning unit: a job occupying an absolute position. "Node" here denotes position-to-job evidence, not a graph-vertex model; we make this explicit once, following [24]'s original usage, to avoid confusion with graph-based EDA terminology.

CNB-COIN (Consecutive Node-Based COIN). Uses the identical $W[p, j]$ model and learning update as NB-COIN. The only difference is that positions are filled consecutively from 0 to $n - 1$ rather than in a randomized order. It is an ordering ablation of the node sampler, not a new probability table, and isolates whether position-fill order alone affects Pareto-approximation quality.

SNE-COIN (Start-Node Edge COIN; working name: Start-Node Edge COIN). State: a first-job distribution vector S[first job] plus the directed edge matrix $W[i, j]$ from plain COIN. Sampling: roulette-sample the first job from S, then complete the permutation from the edge weights as in plain COIN. Learning: both the first-job event and the directed edges are rewarded/punished from the same cohorts. This is the representation with the best mean normalized hypervolume in the confirmatory experiment (Section 5). The mechanistic interpretation we offer is that it supplies the one piece of information the plain edge model does not explicitly encode: which job legitimately opens the sequence. This is an evidence-consistent interpretation, not a proof; Section 6 states directly what would be needed to move it from interpretation to demonstrated mechanism. We do not equate SNE-COIN with Tsutsui's tag-node device [13] without a validated implementation-level equivalence, which we have not established.

HNE-COIN (Hybrid Node–Edge COIN; working name: Hybrid Template COIN). Maintains both a position (node) model and an edge model. A complete permutation is first generated from the position model

as a scaffold; position 0 and a scattered random 30–70% of the remaining positions are then preserved from that scaffold, and the unfilled holes are completed using edge probabilities while respecting the jobs already reserved by the scaffold. Both component models are trained from the completed population every generation. This is *not* the same mechanism as EHBSA/WT's punched-parent replacement (Section 3.3) and not the same as direct COIN/WT: HNE-COIN's "template" is a generated scaffold from its own node model, not a punched real parent, and the infill uses learned edge probabilities rather than resampling from the same node histogram.

CNE-COIN (Chain Node–Edge COIN; working name: Hybrid Chain COIN). Also maintains node and edge models together, but combines them link by link rather than scaffold-then-infill: position 0 is sampled from node evidence, and at every later position the sampler randomly selects either node evidence for that position or edge evidence conditioned on the previously placed job. No punched parent template is involved in CNE-COIN.

3.3. Rebuilt-histogram baselines

EHBSA (Edge-Histogram-Based Sampling Algorithm), WO/WT. Rebuilds a directed edge histogram from the top 50% of the population every generation (bias ratio 0.005). EHBSA/WO samples a complete permutation from that histogram with no template. EHBSA/WT instead selects a parent, punches (marks for resampling) 50% of its positions, resamples those positions from the rebuilt edge histogram, and performs one-to-one parent–child replacement [12].

NHBSA (Node-Histogram-Based Sampling Algorithm), WO/WT. Identical rebuild-and-template mechanics, applied to a rebuilt position-to-job histogram instead of an edge histogram [14].

3.4. Direct template ablations

COIN/WT. Applies the EHBSA-style 50% punched-parent, one-to-one-replacement mechanism directly to the incremental Edge COIN model. It is architecturally distinct from HNE-COIN: COIN/WT punches a real sampled parent and resamples from the same incremental edge statistics used to generate it, coupling persistent signed evidence with direct parental inheritance in a way HNE-COIN's scaffold-and-infill design does not.

NB-COIN/WT. Applies the equivalent NHBSA-style punched-parent mechanism to the incremental NB-COIN model.

Both ablations test RQ2 directly: does a template mechanism that helps one representation family (rebuilt histograms) help another (incremental signed learning) when transplanted without modification? Section 5 reports that it does not.

3.5. Excluded and audit-only families

Two further lines of work exist in the same codebase and are deliberately outside this paper's confirmatory ranking. First, GA/NSGA-style evolutionary crossover baselines were run for audit and validation purposes only; because they use a fundamentally different search operator (crossover/mutation rather than explicit distribution estimation), mixing them into the ranking would confound representation with search-engine choice, which is precisely the confound this paper is designed to avoid. Second, a separate, unpublished family named ROSE (Relative Order Sequence Estimator) fits exact-position probabilities together with signed relative-distance statistics between job pairs; it has since been evaluated on a completed generation-400 scalar audit (four seeds, TA001–TA020) and on the tri-objective pilot introduced in Section 5.6, but it is still not part of the twelve-method multi-objective confirmatory table and is not used to support any confirmatory claim in this paper. Third, the ranking- and latent-priority representations named in Section 2 — Generalized Mallows EDA (GM-EDA), Plackett–Luce EDA (PL-EDA), and random-key EDA (RK-EDA) — are, as of this revision, no longer purely hypothetical future work: each has a single-seed exploratory pilot across the scalar, bi-objective, and tri-objective protocols (Section 5.6), but a single seed cannot support any ranking or significance claim, so they remain excluded from every confirmatory table in this paper and are discussed only qualitatively, where relevant, alongside the tri-objective extension. All three exclusions are scope decisions, not evidentiary weaknesses: this paper's confirmatory claims are about the twelve representations defined above, under Pareto and scalar selection, at the stated budgets.

3.6. Summary and naming

Table 3 summarizes what each method learns and samples, and Fig. 1 illustrates the representational distinction schematically for a five-job toy permutation (not a drawn result). Two of the six COIN-family representations carry renamed labels relative to earlier internal notes on this project; Table 4 records the mapping explicitly so that this paper and any earlier draft referring to the old names can be reconciled without ambiguity.

Table 3: Compared EDA representations and confirmatory hyperparameters.

Method	**Learned / sampled structure**	**Update**
COIN	Directed job-to-job edges	signed, incremental
NB-COIN	Position-to-job; random unfilled-position order	signed, incremental
CNB-COIN	Position-to-job; consecutive position order	signed, incremental
SNE-COIN	First-job distribution plus directed edges	signed, incremental
HNE-COIN	Node scaffold (pos. 0 + scattered 30–70%) infilled by edge probabilities	signed, incremental
CNE-COIN	Node or edge evidence chosen link by link	signed, incremental
EHBSA/WO, /WT	Rebuilt directed edge histogram, without / with 50% punched template	rebuilt, top 50%
NHBSA/WO, /WT	Rebuilt position histogram, without / with 50% punched template	rebuilt, top 50%
COIN/WT, NB-COIN/WT	Incremental edge / position models with a 50% punched template grafted on	signed + punched

Table 4: Naming reconciliation between this paper and earlier internal working names.

This paper	**Earlier working name**
COIN	COIN (unchanged)
NB-COIN	NB-COIN (unchanged)
CNB-COIN	Consecutive Node-Based COIN (unchanged)
SNE-COIN	Start-Node Edge COIN
HNE-COIN	Hybrid Template COIN
CNE-COIN	Hybrid Chain COIN
COIN/WT, NB-COIN/WT	COIN/WT, NB-COIN/WT (unchanged)

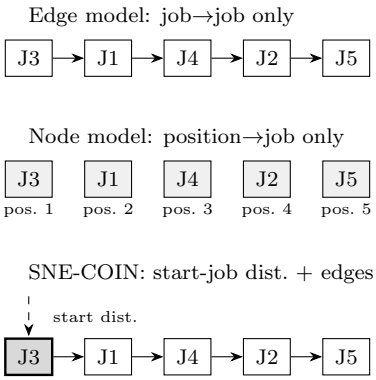

Figure 1: Schematic contrast of three representational families (not a drawn result): the plain edge model does not explicitly encode a legitimate opener; the node model does not explicitly encode adjacency; SNE-COIN retains both an edge model and an explicit first-job distribution.

4. Experimental protocol

4.1. Instances, budget, and seeds

We use TA001–TA030: ten 20×5, ten 20×10, and ten 20×20 instances, so the comparison spans three problem sizes rather than one. Every method uses population 100, 400 generations, and therefore 40,000 evaluations per run, with the cohort and learning-rate hyperparameters given in Section 3. Thirty seeds (42 and a sequential block from 2026 through 2054) are shared across all twelve methods so that per-seed randomness is a paired, not confounding, factor. GA/NSGA-style crossover baselines were run for audit purposes only and are excluded from the principal ranking because they use a different search operator, which would confound representation with search-engine choice (Section 3); the unpublished ROSE family is outside the scope of this paper for the same reason plus its incomplete backfill status.

4.2. Normalization and reference front

For each instance, objective values are normalized once using the ideal and nadir values observed across all twelve methods and all thirty seeds, and that single normalization is fixed for every downstream metric. The pooled observed nondominated set — the union of nondominated points across all twelve methods and all complete seeds, per instance — is used as the reference set. It is a pooled observed reference front, not an assertion of the true Pareto front, since no exact PFSP Pareto front is available at this scale. We flag explicitly that this construction is self-referential: the reference front is built from the same twelve methods whose HV, IGD+, and pooled-front contribution are then measured against it. A method that occupies a locally unique region of objective space mechanically contributes more of the reference front and can move its own HV/IGD+ favorably as a result, independent of any property beyond occupying that region first. We do not believe this artifact explains the magnitude of the gaps reported in Section 5 — the leading method's advantage survives Holm-corrected pairwise testing against every competitor — but it is a limitation of any pooled-front design, including this one, and a fully independent (e.g., exact or externally sourced) reference front is not available for these instances.

4.3. Indicators

Hypervolume (HV) uses reference point $(1.1, 1.1)$ in normalized space, larger is better: for an approximation set A and reference point r,

$$\mathrm{HV}(A, r) = \Lambda\left(\bigcup_{a \in A} [a_1, r_1] \times [a_2, r_2]\right), \tag{1}$$

the Lebesgue measure Λ of the region jointly dominated by A and bounded above by r. IGD+ measures the distance from the pooled reference front R to a method's approximation A and is reported instead of ordinary IGD because it does not penalize points in A that already dominate a reference point [27]:

$$\mathrm{IGD}^{+}(A, R) = \frac{1}{|R|} \sum_{r \in R} \min_{a \in A} d^{+}(a, r), \tag{2}$$

with $d^{+}(a, r) = \sqrt{\sum_{i=1}^{2} \max(a_i - r_i, 0)^2}$ for the two minimized, normalized objectives; smaller IGD+ is better. We additionally report each method's contribution share (its portion of the pooled front), final archive size (capped only by dominance, never by a size parameter), and wall-clock runtime. Runtime, in both selection regimes studied in this paper, is implementation- and hardware-dependent and reported only for relative comparison among methods run on identical hardware, never as an absolute performance claim; we also do not merge it into any objective-quality ranking or score, because a method that is fast and a method that is accurate are answering different questions, and collapsing them into one number would hide exactly the trade-off a reader needs to see (this matters concretely for the single-objective runtime column reported in Section 5.5, where the fastest and most accurate methods are not the same method).

4.4. Single-objective confirmatory protocol

The matched scalar experiment (Section 5.5) uses TA001–TA020 (the 20×5 and 20×10 Taillard subset used elsewhere in this paper, restricted to the first twenty instances), four paired seeds (42, 2026, 2027, 2028, a subset of the thirty multi-objective seeds), population 100, and 400 generations, for the same twelve methods defined in Section 3, run separately for makespan and for total flow time. Every run therefore uses the identical 40,000-evaluation budget as the multi-objective experiment; only the selection rule changes, from Pareto dominance plus crowding to a single scalar objective. This gives

$20 \times 4 = 80$ paired instance–seed blocks per objective, and $80 \times 12 \times 2 = 1{,}920$ completed runs in total, with no recorded run failures. Convergence is plotted, and final performance is tabulated, as *average relative percentage deviation* (ARPD) from the best final objective value observed among the twelve compared methods, computed within each instance–seed block and then averaged across the 80 blocks:

$$\mathrm{ARPD}_{i,s,a} = 100 \times \frac{f_{i,s,a} - f^*_{i,s}}{f^*_{i,s}}, \qquad f^*_{i,s} = \min_{a' \in \mathcal{A}} f_{i,s,a'}, \tag{3}$$

where $f_{i,s,a}$ is method a's final objective value on instance i, seed s, and $\mathcal{A}$ is the set of twelve compared methods; lower ARPD is better, and $\mathrm{ARPD} = 0$ marks the block's best observed method. We emphasize what this normalization is and is not: $f^*_{i,s}$ is the best value *observed among the twelve compared methods* for that instance–seed block, not a published optimum or a lower bound, so ARPD here measures relative standing within this comparison, not distance from a known-optimal schedule. Reported confidence intervals are 95% normal-approximation intervals, $\mathrm{CI}_{95} = 1.96 \times \mathrm{SE}$ with $\mathrm{SE} = \mathrm{SD}/\sqrt{n}$ and $n = 80$ blocks; standard deviation, standard error, and the 95% confidence half-width are three different quantities and are not used interchangeably anywhere in this paper. Mean rank is the average, over the same 80 blocks, of each method's competition rank (ties receive the mean of the tied ranks) among the twelve methods on that block, with rank 1 best. Runtime is each run's recorded wall-clock time, reported separately from, and never blended into, the objective-quality ranking, for the reasons given in Section 4.3. As in the multi-objective protocol, GA/NSGA-style crossover baselines and the ROSE family are excluded from this comparison; ROSE's completed scalar batch additionally stored its per-generation history as display strings rather than structured records, which supports final objective values and runtime but not a reconstructed convergence trajectory, so ROSE is not drawn in the convergence figures and is not part of this paper's confirmatory ranking in either selection regime.

4.5. Statistical procedure

Confidence intervals for the multi-objective results use the 900 instance–seed blocks (30 instances $\times$ 30 seeds). A Friedman test compares all twelve methods on HV, followed by paired Wilcoxon signed-rank tests against the top method with Holm correction [28]. The block structure is chosen deliberately: an instance–seed pair, not a method, defines a block, so every

method is compared against every other method on the *same* 900 problem–seed combinations rather than on independently drawn samples; this is what licenses the paired Wilcoxon test in place of an unpaired alternative, and it is also why runtime and hardware variation across blocks cancels out of the paired HV difference even though it does not cancel out of the raw runtime column. The Friedman test is the omnibus check that not all twelve methods are interchangeable; it does not by itself identify which pairs differ. Holm correction is applied to the twelve pairwise tests against the leading method to keep the family-wise error rate bounded without the overcautious conservatism of a flat Bonferroni correction across all $\binom{12}{2} = 66$ possible pairs, most of which are not the comparison of interest here. We report median paired HV differences as the effect-size measure available from the supplied analysis; a standardized rank-based effect size (e.g., matched rank-biserial correlation) computed from the underlying per-block values is identified as a recommended addition in Section 8 rather than estimated indirectly from the summary statistics reported here.

4.6. Tri-objective protocol and internal machine idle time

We extend the protocol above with a third objective, internal machine idle time, defined for machine $k = 1, \ldots, m$ and permutation π as

$$I_{\text{internal}}(\pi) = \sum_{k=1}^{m} \left[C_{k,n}(\pi) - S_{k,1}(\pi) - \sum_{j=1}^{n} p_{\pi_j,k} \right], \tag{4}$$

where $C_{k,n}(\pi)$ is the completion time of the last job on machine k, $S_{k,1}(\pi)$ is the start time of the first job on machine k, and $p_{\pi_j,k}$ is the processing time of the job in position j on machine k. Equation 4 excludes idle time before the first processed job and after the final completion on each machine — it measures only the gaps a schedule inserts *between* jobs it has already started processing on that machine. This is deliberately different from total horizon idle time,

$$mC_{\max} - \sum_{j,k} p_{j,k}, \tag{5}$$

which is an affine transformation of makespan ($C_{\max}$) alone and therefore cannot serve as an independent objective: minimizing Equation 5 is arithmetically equivalent to minimizing $C_{\max}$ for a fixed instance, since $\sum_{j,k} p_{j,k}$ and m are constants of the instance, not of the schedule. Equation 4 has no

such redundancy: two permutations with identical makespan can have different internal idle time, because internal idle time depends on *where*, not only *how much*, machines wait between jobs. We verified this non-redundancy is not merely a formal distinction by confirming that the internal-idle and makespan objective values are not perfectly rank-correlated within the completed tri-objective pilot runs described below (a full correlation analysis is deferred to Section 8 as future confirmatory work; the pilot data are sufficient only to rule out a trivial affine relationship, not to characterize the correlation's strength).

The tri-objective extension uses the same population (100), generation budget (400), and Taillard instance range (TA001–TA020) as the scalar and bi-objective protocols above, jointly minimizing makespan, total flow time, and internal machine idle time under the same Pareto-depth-plus- crowding selection rule described in Section 4, with no scalarization. Its evidentiary status is deliberately reported here, before any result is discussed, rather than only in the limitations section: a fully completed three-seed pilot (seeds 42, 2026, 2027) exists across the twelve core representations plus the ROSE/WT template ablation (thirteen methods, 780 runs, no recorded failures), and a separate, larger batch targeting a ten-seed exploratory checkpoint had reached eight complete seeds (one of the eight itself incomplete) at the time of writing, additionally including the three exploratory representations named in Section 3 (GM-EDA, RK-EDA, PL-EDA) evaluated so far at a single seed only. Neither batch approaches the thirty-seed standard of this paper's bi-objective confirmatory result, and neither has yet had a hypervolume, IGD+, or Friedman/Holm statistical pipeline applied to it as of this writing. We report this status explicitly in Table 5 so that the reader encounters the evidentiary tier before, not after, the exploratory pattern discussed in Section 5.6.

Table 5: Evidentiary status of the tri-objective extension (makespan + total flow time + internal machine idle time) at the time of writing. Contrast with the thirty-seed, 900-block bi-objective confirmatory result in Table 6.

Batch	**Seeds**	**Methods**	**Status**
Small pilot	3 (complete)	12 core + ROSE/WT (13)	Complete; no indicator computed yet
Larger batch	8 of 10 target (1 incomplete)	12 core + ROSE/WT + GM/RK/PL-EDA (16)	In progress; no indicator computed yet
Journal-confirmatory target	20	(to be finalized)	Not started at this scale

5. Results

5.1. Confirmatory ranking

Table 6 reports the confirmatory outcome over all 10,800 runs. SNE-COIN has the highest mean HV and the best HV rank; NB-COIN has the lowest IGD+. No single method wins on every indicator, but the ordering is not scattered: the top three methods by HV — SNE-COIN, HNE-COIN, and NB-COIN — are also the three that combine node- or start-level evidence with edge evidence, and the IGD+ rank ordering tracks the HV ordering closely for these three.

Contribution shares are small in absolute terms (below 1.3% for every method) because the pooled front is assembled from twelve methods and thirty seeds at once; read relatively, SNE-COIN's 1.23% share is roughly 1.4× the next-largest contributor, NHBSA/WO (0.86%), and more than ten times the median contributor across the twelve methods. Final archive sizes are compact throughout, from 5.5 points (COIN/WT) to 8.5 points (NHBSA/WT), which is unsurprising for a two-objective, bounded-instance search space rather than a sign of premature convergence specific to any one method.

5.2. Statistical significance

The Friedman statistic is $\chi^2 = 8040.466$ over 900 blocks, with $p < 10^{-300}$: equal HV performance across the twelve methods is rejected outright. SNE-COIN's paired HV advantage remains significant against every one of the

Table 6: Confirmatory results over 10,800 runs (30 seeds × 30 instances × 12 EDAs). Renamed methods carry their earlier working names in parentheses on first use in text.

Method	HV ↑	HV rank ↓	IGD+ ↓	Contribution	Time (s)
SNE-COIN	**0.9871**	**2.90**	0.1149	**1.23%**	15.62
HNE-COIN	0.9808	2.94	0.1117	0.23%	25.59
NB-COIN	0.9708	3.45	**0.1096**	0.32%	17.95
CNB-COIN	0.9610	3.76	0.1223	0.52%	17.47
CNE-COIN	0.9597	3.83	0.1312	0.12%	18.27
NHBSA/WO	0.9033	5.48	0.1503	0.86%	18.90
NHBSA/WT	0.8866	6.19	0.1865	0.04%	14.80
EHBSA/WT	0.8015	8.19	0.2433	0.01%	14.36
COIN	0.7701	8.69	0.2625	0.04%	**7.92**
NB-COIN/WT	0.6996	10.22	0.3118	0.00%	15.70
EHBSA/WO	0.6696	10.74	0.3401	0.01%	18.18
COIN/WT	0.6205	11.60	0.3734	0.00%	16.50

eleven competitors after Holm correction. Against its closest rival, HNE-COIN, the median paired HV difference is 0.0186 (Wilcoxon statistic 173542, Holm-adjusted $p = 0.000183$); the margin against the weakest method, COIN/WT, is 0.372 (Holm-adjusted $p = 7.4 \times 10^{-148}$). The result is consistent enough across this many paired comparisons that it does not read like a borderline finding, though we reiterate that at $n = 900$ blocks even small, practically unimportant differences will register as statistically significant; the median paired differences reported here, not the p-values alone, are the basis for any practical claim of superiority.

5.3. Template behavior is representation-dependent

Template behavior is representation-dependent rather than uniformly helpful or harmful. EHBSA benefits from the punched template (HV 0.8015 versus 0.6696 without it), but NHBSA/WT falls below NHBSA/WO. More strikingly, grafting the same punched-template mechanism directly onto incremental COIN hurts both hosts: COIN/WT and NB-COIN/WT rank twelfth and tenth, well below their un-templated counterparts. These are negative ablations, and they are reported as results in their own right, because they rule out the tempting generalization that punched templates are a free improvement transferable across any EDA family. Note that HNE-COIN — which also combines node and edge evidence, but through its own

scaffold-and-infill mechanism rather than a punched real parent (Section 3.3) — does not show this degradation, which is consistent with the two mechanisms being genuinely different rather than differently tuned versions of the same idea; we have not, however, isolated the scaffold-construction cost from the edge-infill cost within HNE-COIN, so its runtime relative to SNE-COIN (25.6 s vs. 15.6 s) is reported without a causal decomposition.

5.4. A representative instance-level view

The pooled ranking in Table 6 is the confirmatory evidence; Fig. 2 supplements it with a single representative instance, TA022, showing each algorithm's own union frontier across all 30 seeds against the global pooled front for that instance. This is illustrative of one 20×20 instance and is not itself confirmatory evidence of the pooled ranking; we include it because it makes visible, for one concrete case, what the aggregate ranking states abstractly — namely that the COIN-family frontiers (left panel) sit closer to the global pooled front than the histogram-EDA frontiers (right panel) across most of the trade-off region for this instance.

5.5. Single-objective confirmatory results

Table 7 and Table 8 report the matched single-objective comparison specified in Section 4.4: TA001–TA020, four paired seeds, population 100, and 400 generations, for makespan and total flow time respectively, with ARPD measured against the best final value observed among the twelve compared methods in each instance–seed block (lower is better for ARPD and for mean rank; runtime is reported separately and is not part of the quality ranking). We state the headline finding plainly because it is the point of running this experiment: the representations that lead under Pareto selection (Section 5) do *not* lead here. Punched-template methods occupy the top of both rankings — NHBSA/WT and NB-COIN/WT are first or second on both objectives, and EHBSA/WT is third on total flow time — while SNE-COIN, HNE-COIN, and CNE-COIN, the three top performers under Pareto selection, sit in the middle third of both scalar rankings. COIN and EHBSA/WO rank last on both objectives, by a wide margin.

The runtime column is reported for completeness and is deliberately not folded into the ranking above it: COIN is both the worst method by ARPD and, by a wide margin, the fastest per run on both objectives (0.16 s and 0.09 s mean wall-clock time, versus 3.7–10.0 s for every other method), because its edge-only sampler and update touch far less state per generation

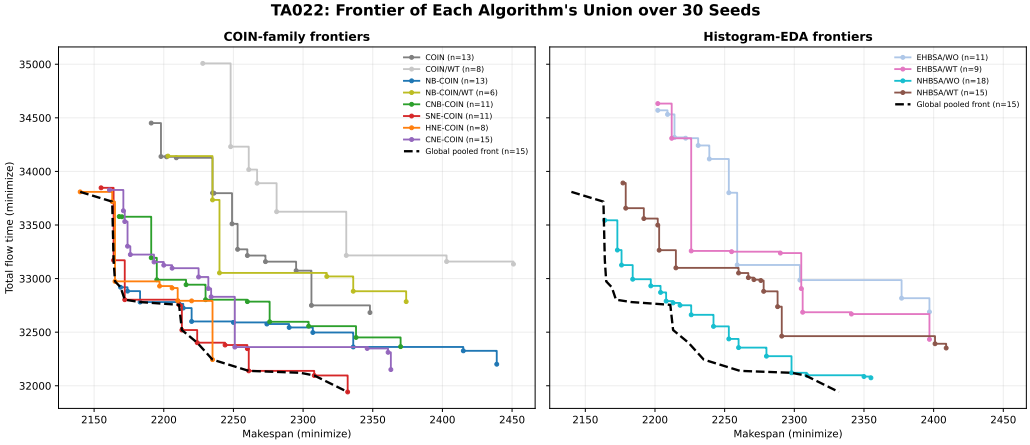

Figure 2: Each algorithm's own union frontier over 30 seeds on instance TA022 (one representative 20×20 instance), split into COIN-family (left) and histogram-EDA (right) panels, against the global pooled observed front for that instance (dashed). Illustrative of one instance; Table 6 is the confirmatory, pooled-instance evidence. **Naming note:** this figure's legend was generated before the naming reconciliation in Table 4 and still reads "Start-Node Edge COIN," "Hybrid Template COIN," and "Hybrid Chain COIN"; read these as SNE-COIN, HNE-COIN, and CNE-COIN respectively. Regenerating this figure under the final naming is a pre-submission task, since the underlying plotting script was not among the files available for this revision — only the rendered figure and its source CSVs were.

than a position-indexed or dual-model sampler. A fast method that finishes far from the block-best value is not a good trade merely by virtue of being fast, and we do not construct a combined speed–quality score from these two columns for that reason. HNE-COIN is the slowest method on both objectives (approximately 10 s per run), which is consistent with its scaffold-then-infill construction touching two full models every generation; as in the multi-objective results (Section 6), we have not decomposed this cost into its scaffold and infill components.

Fig. 3 and Fig. 4 plot the full convergence trajectories underlying Table 7 and Table 8. Between generation 200 and generation 400, every one of the twelve methods continues to decrease in mean ARPD on both objectives (Section 4.4 checkpoint data; no method reaches a literally flat trajectory), so 400 generations does not represent full convergence for any method in the strict sense of zero further improvement. The rate of improvement is not uniform, however. The leading WT-templated methods (NHBSA/WT, NB-

Table 7: Single-objective confirmatory results for makespan, TA001–TA020, 4 seeds, population 100, 400 generations (80 paired instance–seed blocks). ARPD is the mean percentage deviation from the best final value observed among these twelve methods in each block (lower is better); 95% CI is the normal-approximation half-width ($1.96 \times$ SE, $n = 80$). Mean rank (lower is better) is the average competition rank across the same 80 blocks. Runtime is mean wall-clock seconds ± SD per run on identical hardware, reported separately from the quality ranking (Section 4.3). Bold marks the best value in each column; computed reproducibly from the raw per-run JSON archives (Section 4.4).

Method	**ARPD % (95% CI)**	**Mean rank**	**Runtime (s)**
NHBSA/WT	**0.32 (0.13)**	**2.57**	3.74 (0.57)
NB-COIN/WT	0.47 (0.16)	2.76	4.70 (0.65)
NB-COIN	1.08 (0.23)	4.24	6.13 (0.92)
CNB-COIN	1.48 (0.27)	5.51	5.60 (0.88)
EHBSA/WT	1.49 (0.22)	5.74	4.01 (0.67)
HNE-COIN	1.56 (0.24)	5.93	10.02 (1.37)
COIN/WT	1.81 (0.26)	6.38	4.92 (0.68)
CNE-COIN	2.07 (0.29)	7.26	6.47 (0.83)
SNE-COIN	2.31 (0.32)	7.72	4.46 (0.65)
NHBSA/WO	2.78 (0.42)	8.60	5.80 (0.90)
COIN	3.81 (0.45)	10.14	**0.16 (0.47)**
EHBSA/WO	4.57 (0.48)	11.16	5.64 (0.88)

COIN/WT, EHBSA/WT) approach an ARPD floor below 1–2% by generation 250–300 and their generation 300-to-400 improvement is small in absolute terms (0.2–0.6 percentage points on both objectives), consistent with the visually near-flat tail of their curves in Figs. 3–4. COIN and EHBSA/WO, by contrast, remain several percentage points from the block-best value at generation 400 and their generation 300-to-400 improvement is comparable in absolute size to (and, for COIN, larger than) the improvement shown by the leading methods over the same interval, which is evidence that these two methods are still improving at generation 400 rather than having plateaued at a worse level; whether they would eventually reach the WT methods' final ARPD under a longer budget is not something this experiment tests; and we do not claim it either way.

The two objectives largely agree on which methods lead and which trail, but not on the fine ordering in between, which is itself informative: NHBSA/WT and NB-COIN/WT swap first and second place between makespan and total flow time, EHBSA/WT moves from fifth (makespan) to third (total flow

Table 8: Single-objective confirmatory results for total flow time, TA001–TA020, 4 seeds, population 100, 400 generations (80 paired instance–seed blocks). Columns and normalization as in Table 7.

Method	**ARPD % (95% CI)**	**Mean rank**	**Runtime (s)**
NB-COIN/WT	**0.29 (0.10)**	**2.03**	4.53 (0.63)
NHBSA/WT	0.52 (0.14)	2.76	3.72 (0.53)
EHBSA/WT	0.88 (0.16)	3.66	3.95 (0.56)
CNB-COIN	1.35 (0.18)	5.29	5.62 (0.75)
COIN/WT	1.39 (0.18)	5.51	4.73 (0.63)
HNE-COIN	1.64 (0.17)	6.29	9.98 (1.18)
NB-COIN	1.64 (0.17)	6.46	6.10 (0.77)
CNE-COIN	1.89 (0.22)	6.96	6.45 (0.77)
SNE-COIN	2.12 (0.29)	7.32	4.52 (0.60)
NHBSA/WO	2.74 (0.28)	8.76	5.73 (0.77)
COIN	5.31 (0.35)	11.08	**0.09 (0.01)**
EHBSA/WO	6.75 (0.40)	11.88	5.59 (0.69)

time), and HNE-COIN and NB-COIN are close enough on total flow time (1.64 vs. 1.64, mean rank 6.29 vs. 6.46) to be effectively tied given their overlapping confidence intervals, whereas on makespan NB-COIN is clearly ahead of HNE-COIN (1.08 vs. 1.56). Adjacency and positional information therefore do not carry equal value under the two objectives even within this single selection regime, consistent with total flow time's sensitivity to the completion time of every job rather than only the last one (Section 2). A Spearman rank correlation across the twelve methods' mean ranks on the two scalar objectives is $\rho = 0.909$ ($p < 10^{-4}$, $n = 12$ methods): the two scalar objectives agree closely with each other on which representation is preferable. The same correlation computed between each method's multi-objective HV rank (Table 6) and its scalar mean rank is $\rho = 0.147$ ($p = 0.649$) against makespan and $\rho = -0.070$ ($p = 0.829$) against total flow time: statistically indistinguishable from no monotonic relationship at all, and, for total flow time, even weakly in the opposite direction. With only twelve paired methods this correlation test has limited power to detect anything short of a strong relationship, so we do not read the non-significant p-values as proof of zero relationship; what the coefficients do show, at face value, is that whatever relationship exists between the multi-objective and scalar rankings is far weaker than the relationship between the two scalar rankings themselves,

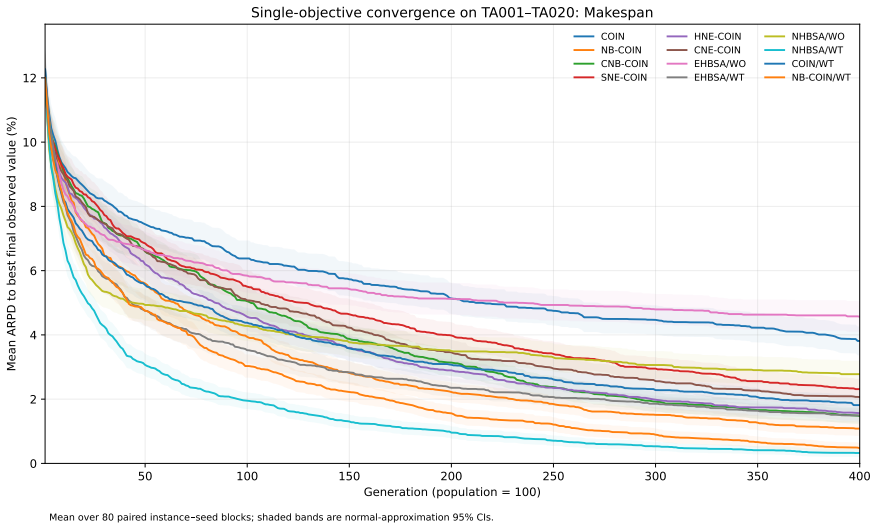

Figure 3: Single-objective convergence, makespan, TA001–TA020, 4 paired seeds, population 100, 400 generations. Each curve is the mean ARPD (%) from the best final makespan value observed among these twelve methods, normalized within each instance–seed block before averaging over the 80 blocks; lower is better. Shaded bands are normal-approximation 95% confidence intervals. ROSE/WT is not drawn (Section 4.4).

which is the quantitative form of this section's headline finding: the Pareto-selection ranking and the scalar-selection ranking are not the same ranking wearing different units.

5.6. Tri-objective exploratory results

We report a preliminary three-objective extension — makespan, total flow time, and internal machine idle time jointly — under the protocol and evidentiary status stated in Section 4.6 and Table 5: a fully completed three-seed pilot and a separate, larger batch at eight of an intended ten-seed exploratory checkpoint (itself short of the twenty-seed target we consider necessary for a confirmatory tri-objective claim), neither of which has yet had a hypervolume, IGD+, or Friedman/Holm significance pipeline run against it.

We have not computed a three-dimensional hypervolume or IGD+ value for this paper, and we do not report one here in place of a genuine computation with a declared reference point and normalization convention matched to the bi-objective protocol of Section 4. **[VERIFY / TO COMPLETE: insert the completed three-seed pilot's HV/IGD+/mean-rank table**

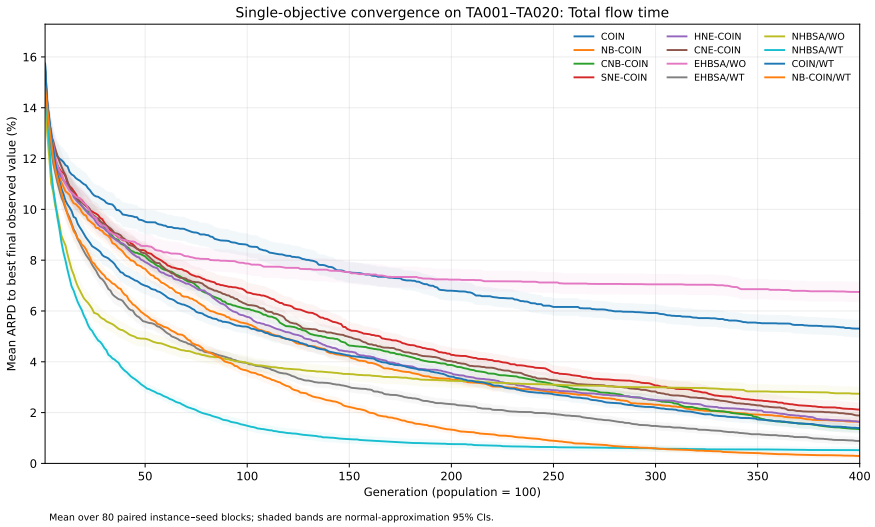

Figure 4: Single-objective convergence, total flow time, TA001–TA020, 4 paired seeds, population 100, 400 generations. Same normalization, block structure, and exclusions as Fig. 3; lower is better.

here once that pipeline has been run against the tri-objective pilot's raw run archive, with an explicitly declared three-dimensional reference point, e.g. $(1.1, 1.1, 1.1)$ **under the same per-instance normalization used for Table 6; until that computation exists, this subsection reports only completion status and a qualitative, explicitly uncomputed-in-this-manuscript observation below, neither of which should be cited as a confirmatory finding of this paper.**]

The pattern we can report without a completed indicator pipeline is limited to what the raw run structure itself supports: the tri-objective experiment ran without recorded failures on every seed it completed, using the same twelve-representation-plus-ROSE/WT set as the bi-objective study for the three-seed pilot, and additionally the three exploratory representations (GM-EDA, RK-EDA, PL-EDA) for the larger, still-incomplete batch. We do not report a ranking, a winning method, or a hypervolume value in this subsection, because none has been computed from these files at the time of this writing. A prior informal inspection of the pilot data suggested a pattern in which NB-COIN shows strong hypervolume and IGD+, HNE-COIN shows a strong mean rank, and NHBSA/WO contributes pooled-front regions distinct from the COIN-family methods; we relay this as an author-supplied,

uncomputed-in-this-manuscript observation for the reader's context, explicitly not as a result of this paper's own analysis pipeline, and it must not be cited as this paper's finding until the indicator computation described above has actually been run and its output has replaced this paragraph.

6. Discussion

SNE-COIN's advantage is consistent with the interpretation that it supplies information the plain edge model does not explicitly encode: directed edges describe transitions between jobs but say nothing about which job may legitimately open the permutation, so the plain edge model must approximate that choice uniformly at random every generation. Adding an explicit, learned first-job distribution removes that gap. We state this as an interpretation consistent with the ranking, not as a mechanism we have directly measured; doing so would require tracking model entropy or first-job concentration over generations, which is not part of the current instrumentation (Section 8).

HNE-COIN covers slightly less hypervolume but reaches competitive IGD+ at a materially higher runtime. One plausible reading is that it spends extra computation on front coverage that SNE-COIN obtains more directly; an equally plausible reading is that the scaffold-construction step alone accounts for the runtime gap independent of any coverage benefit. We present both readings because the current data — aggregate runtime only — cannot distinguish them (Section 7, threat of confounded mechanisms). NB-COIN's lead on IGD+ despite a smaller dominated area indicates that its approximation sits close to the pooled reference front on average, even where it does not push the front's extremes as far out; this is a direct, low-inference reading of what IGD+ measures rather than a claim about NB-COIN's internal mechanism.

The punched-template ablations show that the same mechanism can help one model family and harm another: it improves a from-scratch histogram model (EHBSA) while destabilizing an incremental, signed-cohort model (COIN, NB-COIN), most plausibly because half-template replacement conflicts with the incremental model's implicit assumption that reward and punishment statistics evolve smoothly between generations rather than being partially overwritten by a punched parent copy. This is again an interpretation rather than a demonstrated mechanism; distinguishing "conflicts with smooth evidence accumulation" from other candidate explanations (e.g.,

reduced effective exploration under template constraint) would require the same entropy/diversity instrumentation called for above.

Taken together, the pattern does not reduce to "COIN beats histograms" or "templates help." It is closer to: representations that make more of the permutation's structure explicit (start position, or position and edge jointly) tend to do better under Pareto selection at this budget, and a template mechanism's value depends on whether the host model rebuilds its statistics from scratch (where a template supplies useful structure a from-scratch model would otherwise lack) or maintains them incrementally (where the same template can overwrite structure the model had already learned).

6.1. A general limitation of multi-objective EDAs

These findings should also be interpreted against a general limitation of multi-objective EDAs. A non-dominated population need not represent a single coherent distribution. Solutions near different regions of a Pareto front may embody conflicting permutation structures: an edge or position that supports makespan may be unfavorable for total flow time, while extreme solutions may be represented by relatively few samples. Estimating one global model from this population can therefore average incompatible modes, dilute objective-specific dependencies, and generate permutations that correspond well to no particular region of the front. As probabilities become concentrated, this structural averaging may further reduce exploration and make lost modes difficult to recover.

This issue is especially important for low-order histogram models. Edge and node-position matrices capture different first-order regularities, but neither explicitly represents the conditional statement that a structure is useful only in a particular region of objective space. Consequently, accurate estimation of the selected population does not necessarily imply an adequate representation of its multimodal structure. Archive truncation and density control can introduce an additional bias: sparsely sampled extreme regions provide weaker evidence to the model than the densely populated center, potentially improving central convergence while sacrificing coverage.

COIN does not eliminate these limitations. Its signed evidence from rewarded and punished cohorts may retain information that a positive-only rebuilt histogram discards [31], but a single COIN model can still merge incompatible structural modes, become overconfident, and depend strongly on selection rate, smoothing, and representation. The superior multi-objective performance of SNE-COIN and the IGD+ behavior of NB-COIN (Section 5)

are therefore consistent with improved handling of heterogeneous evidence, but they do not establish that negative learning directly preserves decision-space diversity. Confirming that mechanism requires measurements of population entropy, edge and position entropy, pairwise permutation distance, objective-space spread, and representation-specific model concentration over generations (Section 8).

6.2. Does a third objective amplify or dampen the representation effect? (exploratory)

Section 6.1 argues that a nondominated population under two objectives can already contain mutually incompatible permutation modes that a single global low-order model must average over, diluting objective-specific dependencies and underrepresenting extreme regions of the front. A third, non-redundant objective should make this averaging problem harder, not easier: a schedule that trades off makespan and total flow time well is not guaranteed to also manage internal machine idle time well, so the nondominated set a tri-objective run must represent is at least as heterogeneous as the bi-objective set, plausibly more so. We treat this as a testable prediction of the structural-averaging argument already made in Section 6.1, not as a new argument requiring new theory: if that argument is right, the representation-ranking spread observed under three objectives should be at least as large as under two, once a comparable indicator pipeline is run on both. We do not yet have the tri-objective indicator computation needed to test this prediction quantitatively (Section 5.6), so this subsection states the prediction and what would confirm or disconfirm it, rather than claiming the exploratory pilot already does so. We explicitly reject the stronger, unsupported version of this argument — that a third objective *proves* representation matters more — in favor of the weaker, correctly scoped one: the tri-objective extension is a stress test whose result is not yet in hand.

6.3. Reading the two selection regimes together

The single-objective result (Section 5.5) is not a contradiction of the multi-objective result; it is a second, independent observation that narrows what the multi-objective result is allowed to mean. If SNE-COIN, HNE-COIN, and CNE-COIN were simply "better representations" in some regime-independent sense, we would expect them to also lead under scalar selection, and they do not: they are outperformed there by punched-template

methods they beat decisively under Pareto selection. The most parsimonious interpretation is not that Pareto selection directly rewards structurally different permutations, but that maintaining objective-space coverage may indirectly require several distinct permutation structures. The additional start-node and combined node-edge information available to the new COIN variants may help represent some of this heterogeneity. Under scalar selection, in contrast, rapid concentration around a smaller set of structures may be advantageous, which is consistent with the stronger performance of punched-template methods. The observed rank reversal therefore supports a representation-by-selection-regime hypothesis, although decision-space diversity measurements (Section 6.1) are required before attributing it specifically to multimodal preservation. We state this as the interaction the paper's title refers to — problem structure (which selection regime is in force), objective function (makespan versus total flow time, which do not even agree with each other on the fine ordering; Section 5.5), and learned representation (what a model counts and how it retains evidence) — rather than as a claim that any one representation is unconditionally superior. It is also why we resist a one-line summary of this paper as "the new COIN variants win": they win specifically under Pareto selection, at this budget, and the single-objective result is the evidence that this specificity is real and not an artifact of only ever having tested one selection regime.

We note one asymmetry the two experiments share rather than resolve. COIN is the weakest method by solution quality in *both* regimes (last under Pareto selection, last under scalar selection on both objectives) while also being by far the fastest per run under scalar selection (Section 5.5); a plausible reading is that its minimal state (no reward/punish memory beyond one edge matrix, no position or start-node information) is simply an under-powered representation for either selection regime, and the various additions studied in this paper — start-node information, position evidence, punched templates — are each, in their own way, compensating for different parts of that shortfall. We present this as a unifying observation about the baseline rather than a new mechanism, since it follows directly from the two confirmatory tables rather than from additional instrumentation.

7. Limitations and threats to validity

The confirmatory results are relative to TA001–TA030 and the declared protocol; they are not evidence about global optimality, a known true Pareto

front, or state-of-the-art status relative to established multi-objective PFSP solvers such as RIPG [5] or NNMA [9], which this paper does not attempt to match (Section 2). The reference front used throughout is a pooled, observed, per-instance front, and, as stated in Section 4, it is self-referential with respect to the methods being scored; we do not believe this drives the reported ranking, but a fully independent reference front is not available for these instances and this remains an open concern for any comparison built this way. Runtime figures are implementation- and hardware-dependent and should be read as relative orderings among methods run under identical conditions here, not as absolute performance claims. Full archive snapshots were not retained at intermediate generations, so the present data support final-generation HV and IGD+ values but cannot be used to draw genuine HV or IGD+ convergence curves; any such curve would be reconstructed, not measured, and is deliberately omitted.

The single-objective comparison (Section 5.5) is a completed, matched, four-seed experiment on TA001–TA020 for all twelve representations, including the COIN/WT and NB-COIN/WT template ablations; it is not, however, a study of the same statistical weight as the 30-seed, TA001–TA030 multi-objective result, and we do not present it as such. Four seeds per instance is the current confirmatory single-objective sample; it supports the qualitative and rank-level conclusions reported in Section 5.5 (which methods lead, which trail, and that the Pareto-selection leaders are not the scalar-selection leaders), but a 30-seed scalar replication, matching the multi-objective sample size, would be needed before treating the exact ARPD magnitudes or the narrower rank gaps (e.g., HNE-COIN versus NB-COIN on total flow time, Section 5.5) as more precisely established than the current confidence intervals already state. We report the comparison between regimes strictly in rank and ARPD terms, as specified in Section 4.4, and do not compare raw scalar objective values against normalized multi-objective HV/IGD+ values, which are not on a common scale and are not commensurable. We also did not retain intermediate archive snapshots for the multi-objective runs (previous paragraph), so while the single-objective convergence curves in Section 5.5 are directly measured, no analogous multi-objective HV/IGD+ convergence curve can be constructed for comparison from the current data. The single-objective experiment also covers a narrower instance range than the multi-objective one: TA001–TA020 spans the 20×5 and 20×10 Taillard subsets but not the 20×20 subset (TA021–TA030) included in the multi-objective comparison, so the RQ3 answer given in Section 5.5 is established

for two of the three problem sizes studied under Pareto selection, not all three; extending the scalar backfill to TA021–TA030 is listed as a concrete next step in Section 8.

The confirmatory statistics reported here are pooled across all three Taillard instance sizes (20×5, 20×10, 20×20); a separate breakdown by size was not part of this analysis, so a size-dependent reversal of the ranking, while not suggested by the pooled result, has not been separately ruled out. The mechanistic interpretations offered in Section 6 (why SNE-COIN wins, why templates conflict with incremental learning) are consistent with the ranking but are not independently measured via model entropy, diversity, or first-job concentration; they should be read as motivated hypotheses. Finally, the twelve-method ranking excludes GA/NSGA-style crossover baselines and the ROSE family by design, as stated in Section 3; both were treated as audit or exploratory work rather than confirmatory evidence in this study's scope, and neither is used to support any claim in this paper.

Tri-objective evidentiary gap. The tri-objective extension reported in Section 5.6 does not meet this paper's own evidentiary bar for the bi-objective and scalar results: it currently has three fully completed seeds (versus thirty for the bi-objective confirmatory result) plus a separate, larger batch at eight of an intended ten-seed exploratory checkpoint (itself short of the twenty-seed target we consider necessary for a confirmatory tri-objective claim), and no hypervolume, IGD+, or significance-test pipeline has yet been run against either batch. We have deliberately not computed or estimated a three-objective indicator value for this manuscript in place of a genuine run of that pipeline, and the qualitative pattern relayed in Section 5.6 is explicitly attributed as an uncomputed, informal prior observation rather than this paper's own result. Any reader treating Section 5.6 as equivalent in evidentiary weight to Sections 5–5.5 is misreading this paper; we have structured the section ordering and the language throughout to make that distinction visible rather than requiring the reader to infer it from a methods table alone. Separately, the internal-machine-idle-time objective's non-redundancy with makespan was checked only for absence of a trivial affine relationship (Section 4.6), not for the strength or shape of whatever weaker association may exist between the two objectives; a full correlation and mutual-information analysis across the completed pilot runs is future work, not a result reported here.

8. Future work

Three lines of work follow directly from the limitations above, and we state them as a concrete plan rather than a vague gesture at "more experiments."

Extending and strengthening the scalar-regime comparison. The current single-objective result (Section 5.5) is confirmatory but narrower than its multi-objective counterpart in two specific ways, and both are addressable without changing the experimental design: (i) a scalar backfill on TA021–TA030 (the 20×20 subset) would extend RQ3's answer to all three instance sizes studied under Pareto selection, not just 20×5 and 20×10; and (ii) increasing the scalar seed count from four to thirty, matching the multi-objective sample, would tighten the confidence intervals in Table 7 and Table 8 enough to resolve the currently-overlapping cases noted in Section 5.5 (e.g., HNE-COIN versus NB-COIN on total flow time) and would support a full Friedman/Wilcoxon significance test analogous to Section 5.2, which the current four-seed sample is not intended to carry.

Instrumenting the mechanism. The interpretations in Section 6 (explicit start-position information removing a recovery burden; punched templates conflicting with incrementally accumulated evidence) motivate measuring model entropy, first-job concentration, and population diversity over generations, at least for a subset of instances and seeds. This would convert the paper's central interpretive claims from consistent-with-the-data to directly measured.

Robustness checks. An instance-size-stratified ($20\times5/20\times10/20\times20$) breakdown of the confirmatory ranking, a machine-count interaction check, standardized rank-based effect sizes (e.g., matched rank-biserial correlation) alongside the median paired differences already reported, and a mixed-effects model with algorithm, instance, and seed as factors, as a cross-check on the Friedman/Wilcoxon result. A runtime decomposition separating HNE-COIN's scaffold-construction cost from its edge-infill cost would resolve the open confound noted in Section 6.

Testing alternative permutation representations. As Section 2 makes explicit, Generalized Mallows EDA (GM-EDA), Plackett-Luce EDA (PL-EDA), and random-key EDA (RK-EDA) encode a permutation through global ranking/ordering statistics or a continuous latent priority rather than through the pairwise-edge or positional-node counts this paper's COIN and histogram families share, and were left out of the present twelve-method con-

firmatory comparison for that reason. This is no longer purely a proposal: each of the three now has a single-seed exploratory pilot across the scalar, bi-objective, and tri-objective protocols (Section 5.6), confirming the harnesses run correctly end to end for all three, but a single seed cannot support any ranking or significance claim. The concrete next step is therefore not "add these representations" but "extend their existing single-seed pilots to the same seed count already used elsewhere in this paper" — four seeds to match the scalar confirmatory standard, thirty to match the bi-objective standard — under the same controlled, equal-budget protocol used throughout. This would test whether the representation × objective × selection-regime interaction we report is specific to edge/node counting models or extends to this structurally different class of representations. The Mallows-kernel construction in particular remains a direct test of the structural-averaging limitation raised in Section 6.1, since it was proposed specifically to escape the single-consensus-permutation assumption that pairwise/positional histogram models share; it has no pilot data yet and remains a from-scratch addition, unlike GM-EDA, PL-EDA, and RK-EDA.

Completing the tri-objective extension. Section 5.6 reports a three-seed completed pilot and a partially completed eight-of-ten-seed exploratory batch for the tri-objective extension (makespan, total flow time, internal machine idle time); neither has a computed hypervolume, IGD+, or significance result. The concrete next steps, in order, are: (i) run the same HV/IGD+ pipeline used for the bi-objective confirmatory result (Section 4) against the completed three-seed pilot, with an explicitly declared three-dimensional reference point and the same per-instance normalization convention; (ii) complete the larger batch to its ten-seed checkpoint and then extend it to the twenty-seed target we consider necessary for a confirmatory tri-objective claim, applying the Friedman-plus-Holm-corrected-Wilcoxon procedure of Section 4 once that count is reached; and (iii) run the correlation/mutual-information analysis between internal machine idle time and makespan flagged as unaddressed in Section 7. Only after these three steps should the tri-objective extension be reported with the same confirmatory language used elsewhere in this paper.

The ROSE family and GA/NSGA/SPEA2 audits are large enough to support their own research questions and are intentionally left to separate future work rather than folded into this paper's scope.

9. Conclusion

This study reports 12,720 completed runs (10,800 multi-objective, 1,920 single-objective) showing that probabilistic representation, not just search-operator choice, materially affects permutation flow-shop scheduling — and that it does so differently depending on how the search is selected. Under Pareto selection over makespan and total flow time, SNE-COIN gives the best hypervolume, NB-COIN the best IGD+, and five of six incremental COIN representations outperform the strongest rebuilt-histogram baseline in mean hypervolume, with the leading method's advantage surviving Holm-corrected testing against every competitor over 900 instance-seed blocks. Punched-template inheritance helps the rebuilt edge histogram but does not transfer as a universal improvement, and actively harms the two incremental COIN models it is grafted onto without modification — a negative result we consider at least as important as the ranking itself, because it rules out a plausible general claim rather than confirming a favored one. Under scalar selection of either objective alone, on a matched four-seed confirmatory sample, this ranking does not carry over: punched-template methods (NHBSA/WT, NB-COIN/WT, EHBSA/WT) lead both objectives, the Pareto-selection leaders (SNE-COIN, HNE-COIN, CNE-COIN) fall to the middle of the scalar ranking, and a Spearman rank correlation between the multi-objective and scalar rankings ($\rho = 0.147$ against makespan, $\rho = -0.070$ against total flow time, $n = 12$) is statistically indistinguishable from no relationship, in contrast to the strong agreement between the two scalar objectives with each other ($\rho = 0.909$). We frame these findings through a three-part structure — objective structure, learned representation, and selection regime — and we observed a clear regime-dependent reversal consistent with an interaction among objective structure, learned representation, and selection regime; establishing the underlying causal mechanism remains a subject for instrumented analysis (Section 6.1, Section 8). Which representation is best is not a fixed property of the representation on this evidence, but depends on whether the search is asked to converge on one scalar winner or to preserve a spread of non-dominated trade-offs. Stated as a single, testable claim: what a probabilistic model over permutations is allowed to represent, whether that representation is rebuilt from scratch or accumulated incrementally, and whether the selection rule rewards convergence or trade-off preservation, jointly shape performance on makespan and total flow time as much as which estimation-of-distribution algorithm's name is on the method

— and it is a claim worth testing in other permutation-based estimation-of-distribution settings, at larger scalar seed counts and across all three instance sizes (Section 8), before it is assumed to generalize beyond this benchmark.

The tri-objective extension of this study (Section 5.6) is a motivating direction, not a third confirmed result standing alongside the bi-objective and scalar findings above. We have verified that internal machine idle time is a genuinely non-redundant third objective, distinct from an affine restatement of makespan, and we have run a substantial fraction of the infrastructure a confirmatory tri-objective comparison would need — a completed three-seed pilot and a larger batch approaching, but not yet reaching, our intended twenty-seed exploratory target. What remains is completing that run and applying the same hypervolume, IGD+, and Holm-corrected significance pipeline already validated on the bi-objective result (Section 8). We report this explicitly as future confirmatory work rather than inflating the current exploratory pattern into a claim this manuscript has not yet earned.

Data and code availability

Configurations, seeds, raw archives, the full literature-evidence matrix, analysis scripts, and versioned source artifacts will be provided with the anonymized submission, subject to the venue's double-anonymous policy.

Supplementary material

The full literature evidence matrix (all reviewed studies, not only those cited in Table 2) and the complete bibliographic verification log (verified/partial status per entry) are retained as supplementary material rather than reproduced in full here, per the review protocol used to compile them. The tri-objective extension's raw per-run results (Section 5.6) are likewise supplementary rather than main-text material at their current evidentiary stage: the completed three-seed pilot and the partially completed eight-of-ten-seed exploratory batch are each retained as separate, explicitly labeled directories rather than merged into a single dataset, since they differ in seed count, completion status, and algorithm coverage (the larger batch alone includes the GM-EDA, RK-EDA, and PL-EDA single-seed pilots). The two secondary bi-objective pairs available from the same internal-idle-time pilot (makespan + internal idle; total flow time + internal idle) are retained as sensitivity-analysis material rather than promoted to main-text status, per the same evidentiary-tier reasoning.

References

[1] E. Taillard, "Benchmarks for basic scheduling problems," *European Journal of Operational Research*, vol. 64, no. 2, pp. 278–285, 1993. doi:10.1016/0377-2217(93)90182-M.

[2] T. K. Varadharajan and C. Rajendran, "A multi-objective simulated-annealing algorithm for scheduling in flowshops to minimize makespan and total flowtime," *European Journal of Operational Research*, vol. 167, no. 3, pp. 772–795, 2005. doi:10.1016/j.ejor.2004.07.020.

[3] J. M. Framinan and R. Leisten, "A multi-objective iterated greedy search for flowshop scheduling with makespan and flowtime criteria," *OR Spectrum*, vol. 30, no. 4, pp. 787–804, 2008. doi:10.1007/s00291-007-0098-z.

[4] G. Minella, R. Ruiz, and M. Ciavotta, "A review and evaluation of multiobjective algorithms for the flowshop scheduling problem," *INFORMS Journal on Computing*, vol. 20, no. 3, pp. 451–471, 2008. doi:10.1287/ijoc.1070.0258.

[5] G. Minella, R. Ruiz, and M. Ciavotta, "Restarted iterated Pareto greedy algorithm for multi-objective flowshop scheduling problems," *Computers & Operations Research*, vol. 38, no. 11, pp. 1521–1533, 2011. doi:10.1016/j.cor.2011.01.010.

[6] T. Pasupathy, C. Rajendran, and R. K. Suresh, "A multi-objective genetic algorithm for scheduling in flow shops to minimize the makespan and total flow time," *The International Journal of Advanced Manufacturing Technology*, vol. 27, nos. 7–8, pp. 804–815, 2006. [**VERIFY: DOI**]

[7] B. Yagmahan and M. M. Yenisey, "Ant colony optimization for multi-objective flow shop scheduling problem," *Computers & Industrial Engineering*, vol. 54, no. 3, pp. 411–420, 2008. doi:10.1016/j.cie.2007.08.003.

[8] S.-W. Lin and K.-C. Ying, "Minimizing makespan and total flowtime in permutation flowshops by a bi-objective multi-start simulated-annealing algorithm," *Computers & Operations Research*, vol. 40, no. 6, pp. 1625–1647, 2013. doi:10.1016/j.cor.2011.08.009.

[9] T.-C. Chiang, H.-C. Cheng, and L.-C. Fu, "NNMA: An effective memetic algorithm for solving multiobjective permutation flow shop scheduling problems," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5986–5999, 2011. doi:10.1016/j.eswa.2010.11.022.

[10] M. M. Yenisey and B. Yagmahan, "Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends," *Omega*, vol. 45, pp. 119–135, 2014. doi:10.1016/j.omega.2013.07.004.

[11] S. Tsutsui, "Probabilistic model-building genetic algorithms in permutation representation domain using edge histogram," in *Proc. Parallel Problem Solving from Nature—PPSN VII*, 2002, pp. 224–233. (Sequence work, not a flow-shop experiment.)

[12] S. Tsutsui and M. Miki, "Solving flow shop scheduling problems with probabilistic model-building genetic algorithms using edge histograms," in *Proc. 4th Asia-Pacific Conf. Simulated Evolution and Learning (SEAL 2002)*, 2002, pp. 776–780. doi:10.1142/9789812561794_0013.

[13] S. Tsutsui, "The effect of introducing a tag node in solving scheduling problems using edge histogram based sampling algorithms," in *Proc. IEEE Congress on Evolutionary Computation*, 2003. **[VERIFY: pages and DOI]**

[14] S. Tsutsui, M. Pelikan, and D. E. Goldberg, "Node histogram vs. edge histogram: A comparison of probabilistic model-building genetic algorithms in permutation domains," in *Proc. IEEE Congress on Evolutionary Computation*, 2006, pp. 1939–1946. doi:10.1109/CEC.2006.1688444. **[VERIFY once more against IEEE Xplore directly; secondary indexes can corrupt this DOI]**

[15] B. Jarboui, M. Eddaly, and P. Siarry, "An estimation of distribution algorithm for minimizing the total flowtime in permutation flowshop scheduling problems," *Computers & Operations Research*, 2009. doi:10.1016/j.cor.2008.11.004. **[VERIFY: volume/pages]**

[16] "Estimation of distribution algorithm for permutation flow shops with total flowtime minimization," *Computers & Industrial Engineering*,

vol. 60, no. 4, pp. 706–718, 2011. doi:10.1016/j.cie.2011.01.005. [**VERIFY: author list from publisher export**]

[17] S. Lemtenneche, A. Bensayah, and A. Cheriet, "An estimation of distribution algorithm for permutation flow-shop scheduling problem," *Systems*, vol. 11, no. 8, Art. no. 389, 2023. doi:10.3390/systems11080389.

[18] J. Ceberio, A. Mendiburu, and J. A. Lozano, "Introducing the Mallows model on estimation of distribution algorithms," in *Proc. 18th Int. Conf. Neural Information Processing (ICONIP)*, 2011, pp. 461–470. doi:10.1007/978-3-642-24958-7_54. (Introduces the Generalized Mallows model as a permutation EDA; the distance-based ranking representation described in Section 2 as GM-EDA.)

[19] J. Ceberio, E. Irurozki, A. Mendiburu, and J. A. Lozano, "A distance-based ranking model estimation of distribution algorithm for the flowshop scheduling problem," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 286–300, 2014. doi:10.1109/TEVC.2013.2260548. (Applies the Generalized Mallows EDA directly to permutation flow-shop scheduling; the GM-EDA comparator cited in Section 2.)

[20] J. Ceberio, A. Mendiburu, and J. A. Lozano, "Kernels of Mallows models for solving permutation-based problems," in *Proc. Genetic and Evolutionary Computation Conference (GECCO)*, 2015, pp. 505–512. doi:10.1145/2739480.2754741. (Proposes a mixture/kernel of Mallows models, each centered on a different reference permutation, to relax the single-consensus-permutation assumption of the plain Mallows model; the Mallows-kernel comparator discussed in Section 6.1. [**VERIFY: it was not possible to confirm from accessible metadata whether this paper's own experiments include permutation flow-shop scheduling specifically, and Section 2 does not assume so**])

[21] J. Ceberio, A. Mendiburu, and J. A. Lozano, "The Plackett-Luce ranking model on permutation-based optimization problems," in *Proc. IEEE Congress on Evolutionary Computation (CEC)*, 2013, pp. 494–501. doi:10.1109/CEC.2013.6557609. (Introduces the Plackett-Luce sequential-selection ranking model as a permutation EDA; the PL-EDA representation cited in Section 2.)

[22] M. Ayodele, J. McCall, and O. Regnier-Coudert, "RK-EDA: A novel random key based estimation of distribution algorithm," in *Proc. Int. Conf. Parallel Problem Solving from Nature (PPSN XIV)*, Lecture Notes in Computer Science, 2016, pp. 849–858. doi:10.1007/978-3-319-45823-6_79. (Introduces the random-key EDA representation: a continuous latent priority per job, sorted to decode a permutation.)

[23] M. Ayodele, J. McCall, O. Regnier-Coudert, and L. Bowie, "A random key based estimation of distribution algorithm for the permutation flowshop scheduling problem," in *Proc. IEEE Congress on Evolutionary Computation (CEC)*, 2017, pp. 2364–2371. doi:10.1109/CEC.2017.7969591. (Applies random-key EDA to permutation flow-shop scheduling specifically; the RK-EDA comparator cited in Section 2.)

[24] O. Srimongkolkul and P. Chongstitvatana, "Application of Node Based Coincidence algorithm for flow shop scheduling problems," in *Proc. 10th Int. Joint Conf. Computer Science and Software Engineering (JCSSE)*, 2013, pp. 49–52. doi:10.1109/JCSSE.2013.6567318.

[25] O. Srimongkolkul and P. Chongstitvatana, "Minimizing makespan using node based coincidence algorithm in the permutation flowshop scheduling problem," in *Industrial Engineering, Management Science and Applications 2015*, M. Gen et al., Eds., Lecture Notes in Electrical Engineering, Springer, 2015, pp. 303–311. doi:10.1007/978-3-662-47200-2_33. (Companion paper to [24], applying the same NB-COIN model to makespan minimization instead of total flow time; both objectives are the ones compared under Pareto selection in this manuscript.)

[26] A. P. Guerreiro, C. M. Fonseca, and L. Paquete, "The hypervolume indicator: Computational problems and algorithms," *ACM Computing Surveys*, vol. 54, no. 6, 2021. **[VERIFY: article number/DOI]**

[27] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *Evolutionary Multi-Criterion Optimization*, 2015. **[VERIFY: LNCS pages/DOI; this is the primary IGD+ definition used throughout this paper]**

[28] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006. (Method adapted here from comparing classifiers over data sets to comparing optimizers over PFSP instance-seed blocks.)

[29] W. Wattanapornprom, P. Olanviwitchai, P. Chutima, and P. Chongstitvatana, "Multi-objective combinatorial optimisation with coincidence algorithm," in *Proc. IEEE Congress on Evolutionary Computation (CEC)*, 2009, pp. 1675–1682. doi:10.1109/CEC.2009.4983143. (Originating paper for the COIN family used throughout this manuscript.)

[30] W. Wattanapornprom, "Hybrid positive and negative correlation learning in estimation of distribution algorithm for combinatorial optimization problems," Ph.D. dissertation, Chulalongkorn University, 2010. (Dissertation; no DOI.)

[31] W. Wattanapornprom and P. Chongstitvatana, "Negative correlation learning in the estimation of distribution algorithms for combinatorial optimization," *IEICE Transactions on Information and Systems*, vol. E96-D, no. 11, pp. 2397–2408, 2013. doi:10.1587/transinf.E96.D.2397. (Formalizes the reward/punish cohort learning rule used by every incremental COIN variant in this paper.)

[32] W. Wattanapornprom and P. Chongstitvatana, "Solving multimodal combinatorial puzzles with edge-based estimation of distribution algorithm," in *Proc. Genetic and Evolutionary Computation Conference Companion (GECCO'11 Companion)*, Dublin, Ireland, 2011. [**VERIFY: page numbers and DOI from the ACM Digital Library record before submission**]

[33] K. Waiyapara, W. Wattanapornprom, and P. Chongstitvatana, "Solving Sudoku puzzles with node based Coincidence algorithm," in *Proc. 10th Int. Joint Conf. Computer Science and Software Engineering (JCSSE)*, 2013. doi:10.1109/JCSSE.2013.6567311. (Cited as evidence that the edge/node representational distinction was already recognized outside PFSP before the present comparison.)

[34] D. E. Goldberg and R. Lingle Jr., "Alleles, loci, and the traveling salesman problem," in *Proc. 1st Int. Conf. Genetic Algorithms and*

Their Applications (ICGA), 1985, pp. 154–159. **[VERIFY: the primary 1985 proceedings text was not directly read in this revision, so the characterization of its schema-theory argument given in Section 1 (that permutation encodings need order/adjacency/position building blocks rather than fixed-locus binary schemata) is a widely repeated secondary characterization of this paper, not a direct quotation from it]**

[35] P. Larrañaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, "Genetic algorithms for the travelling salesman problem: A review of representations and operators," *Artificial Intelligence Review*, vol. 13, no. 2, pp. 129–170, 1999. doi:10.1023/A:1006529012972. (Classifies TSP representations as Binary, Path, Adjacency, Ordinal, and Matrix, and lists path-representation operators including order-based (OX2) and position-based (POS) crossover — the source of the "adjacency"/"ordinal"/"position-based" terminology used in Section 1, reported here using this survey's own category names rather than a paraphrased three-way taxonomy.)

[36] Watcharee Wattanapornprom, T. Li, Warin Wattanapornprom, and P. Chongstitvatana, "Application of node based estimation of distribution algorithms for solving order acceptance with capacity balancing problems by trading off between over capacity and under capacity utilization," in *Proc. 2014 Int. Computer Science and Engineering Conf. (ICSEC)*, 2014, pp. 238–243. doi:10.1109/ICSEC.2014.6978201. Full first names are spelled out here because the manuscript's author (Warin Wattanapornprom, third of four authors) and this paper's first author (Watcharee Wattanapornprom, a different researcher) share a surname and first initial. **[VERIFY: abstract and full text were not accessible in this revision; the paper's specific technical description of any starting-node or hybrid mechanism is therefore not independently confirmed and is not asserted in Section 1 beyond what its title states about its problem domain]**

[37] R. Sirovetnukul, P. Chutima, W. Wattanapornprom, and P. Chongstitvatana, "The effectiveness of hybrid negative correlation learning in evolutionary algorithm for combinatorial optimization problems," in *Proc. 2011 IEEE Int. Conf. Industrial Engineering and Engineering Management (IEEM)*, Singapore, 2011, pp. 476–481.

doi:10.1109/IEEM.2011.6117963. [**VERIFY: abstract and full text were not accessible in this revision; cited only for its title-level connection to the negative-correlation/diversity learning lineage (Section 1), not for any specific technical claim about its internal mechanism**]