applied sciences

MDPI

Article

Reliability-Guided Allocation of Genetic Algorithm Tuning Effort in Permutation Flow Shop Scheduling: A Prospective Validation Study

Nilgün Fığlalı [1,*], Ahmet Cihan [2], Gülşen Aydın Keskin [3], Parmis Şahmaleki Çorbacıoğlu [4] and Alpaslan Fığlalı [1]

[1] Industrial Engineering Department, Engineering Faculty, Kocaeli University, Kocaeli 41001, Türkiye; figlalia@kocaeli.edu.tr
[2] Industrial Engineering Department, Engineering Faculty, Düzce University, Düzce 81620, Türkiye; ahmetcihan@duzce.edu.tr
[3] Industrial Engineering Department, Engineering Faculty, Balıkesir University, Balıkesir 10145, Türkiye; gulsen.keskin@balikesir.edu.tr
[4] Industrial Engineering Department, Engineering Faculty, İstanbul Okan University, İstanbul 34959, Türkiye; parmis.shahmaleki@okan.edu.tr
* Correspondence: figlalin@kocaeli.edu.tr

Featured Application

The framework supports production-scheduling practitioners who already possess calibrated GA parameters and need to allocate tuning effort for a new permutation flow shop problem size. The recommended implementation uses risk-dependent search breadth—narrow axial screening for GREEN, a local Cartesian neighborhood for AMBER, and full-grid screening for RED—followed by finalist confirmation rather than uniform full-replication tuning.

Abstract

Genetic algorithms (GAs) are widely used for permutation flow shop scheduling, but reusing calibrated parameters at a new problem size raises a practical question: how much additional tuning is justified? This study evaluates a reliability-guided rule for allocating that effort. Nine Taillard size groups provide a transferable parameter center and an empirical transfer-reliability index based on held-out same-size instances. For a new size, interpolated reliability determines search breadth and replication effort: GREEN screens the center and available ±1 axial neighbors, AMBER a local Cartesian neighborhood, and RED the full grid, followed by confirmation of leading candidates. The prospective protocol was fixed before outcomes from 24 off-grid targets were observed. At the prespecified 0.25-percentage-point tolerance, both the policy and direct center reuse were within tolerance on 21/24 targets; statistical superiority over direct reuse was not established, although intervention-only analyses favored guided tuning under stricter and continuous-loss summaries. A staged implementation required 3767 tuning runs, versus 30,000 for universal 125 × 10 retuning and 3768 for a reliability-blind budget-matched tuner. Under an interpretive ±0.05% band, fresh validation yielded 9 guided-better, 14 practically tied, and 1 blind-better target. These results support historical transfer reliability as a risk-ranking signal for selective tuning without implying universal superiority.

Keywords: permutation flow shop scheduling; genetic algorithm; algorithm configuration; parameter transfer; tuning effort; transfer reliability; prospective validation

Check for updates

Academic Editors: Arkadiusz Gola and Andrea Prati

Received: 20 September 2026
Revised: 7 October 2026
Accepted: 26 October 2026
Published: 29 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The permutation flow shop scheduling problem (PFSP) is a standard production-scheduling model in which every job visits the machines in the same order and the schedule is represented by a common job permutation. Even this compact formulation becomes computationally difficult as problem size grows, which is why the PFSP remains an important benchmark for exact optimization, constructive heuristics, metaheuristics, and learning-assisted scheduling methods [1–5].

Genetic algorithms (GAs) are attractive for the PFSP because permutations can be manipulated directly by crossover and mutation operators, but their performance depends on configuration choices such as population size, crossover probability, and mutation probability [6–9]. Conventional tuning and automated algorithm-configuration methods address this dependence by searching for good settings under a computational budget [10–13]. The practical difficulty is that a setting selected on one group of problems need not retain the same loss profile on another, while repeating a full configuration search for every new problem can itself become expensive.

Recent research offers two complementary responses to this difficulty. Parameter-control and learning-based approaches adapt settings during a run or learn configuration behavior across problems [14–16]. Transfer-optimization research asks whether knowledge obtained on earlier problems can be reused without causing negative transfer [17]. These developments motivate a different but related decision that arises before a new PFSP is solved: how much target-specific tuning should be performed before previously calibrated parameters are trusted?

This study treats that question as a configuration-effort allocation problem. Historical evidence is organized around nine Taillard size anchors. Each anchor contributes two distinct quantities: a parameter center that serves as the initial recommendation for that size region, and an empirical transfer-reliability index describing how often historically selected configurations remained within a prespecified practical-loss tolerance on held-out same-size instances. The reliability index is therefore evidence about transfer risk, not the success probability of one fixed center and not a calibrated probability for an individual future instance.

For a new size inside the historical envelope, the surrounding anchors are used to interpolate both the parameter center and the reliability index. The center defines where the search begins, while the reliability index determines both search breadth and replication effort. In the recommended staged implementation, GREEN screens the center and available ± 1 axial neighbors, AMBER screens the ± 1 Cartesian neighborhood, and RED retains full 125-configuration coverage; only the leading candidates are then confirmed with additional replications. The colors describe computational action rather than final solution quality.

The evaluation was designed to separate rule construction from rule assessment. Before outcomes from the 24 off-grid prospective targets were observed, the interpolation rule, 0.90/0.75 action thresholds, parameter grid, risk-dependent search regions, conservative prospective tuning schedule, generation-budget rule, stochastic seed families, practical-tolerance criterion, and co-primary statistical tests were fixed in advance. Policy selection, hidden empirical-reference construction, and final validation then used separate stochastic streams. The reduced-replication staged implementation recommended for operational use was evaluated subsequently on the exactly recovered 24-target panel and does not alter the original confirmatory endpoints. In this paper, prospective validation therefore refers to the original temporal separation between development and out-of-sample evaluation; it does not imply universal validity beyond the tested panel.

The analysis distinguishes four questions that can otherwise be conflated: whether historical reliability ranks the risk of direct parameter reuse, whether risk-guided tuning

improves quality when additional tuning is actually invoked, whether historical risk information improves the allocation of a fixed total tuning budget relative to reliability-blind search, and how much configuration work is avoided relative to universal retuning. This separation is important because a policy can be useful for allocating effort even when it does not show statistical superiority over direct reuse on a coarse binary tolerance endpoint.

The intended claim is deliberately bounded. The framework is evaluated as a way to allocate tuning effort while maintaining low observed loss; it is not designed to prove that risk-guided tuning is universally superior to direct reuse. The analyses therefore report direct-center performance, intervention-only AMBER+RED comparisons under the original prospective implementation, surface-flatness controls, computational-work proxies, sensitivity checks, and a budget-matched validation of the final staged implementation against reliability-blind tuning.

The study makes four contributions:

1. First, it formulates GA parameter reuse as a cost–quality decision about how much target-specific configuration evidence to collect, rather than only as a search for a single parameter vector.
2. Second, it separates the transferred parameter center from historical transfer reliability and tests whether the latter carries prospective risk-ranking information on off-grid PFSP sizes.
3. Third, it evaluates the complete rule under a protocol fixed before prospective outcomes were observed with independent tuning, hidden-reference, and final-validation streams and with direct center reuse as an explicit comparator.
4. Fourth, it quantifies the robustness and limits of the decision rule through historical leave-one-instance-out sensitivity, action-threshold sensitivity, a finite-grid random-configuration benchmark for GREEN targets, intervention-only AMBER+RED comparisons, normalized computational-work proxies, and a budget-matched validation showing how the recommended staged implementation reallocates an essentially fixed tuning budget relative to reliability-blind search.

Section 2 positions this decision problem relative to GA parameter setting, automated configuration, adaptive control, instance-specific configuration, and transfer optimization. Section 3 defines the historical anchors, interpolation rule, recommended staged tuning policy, original prospective validation protocol, and budget-matched operational validation. Section 4 reports historical transfer behavior, prospective risk ordering, conservative prospective-policy performance, sensitivity analyses, and the staged policy's matched-budget results. Section 5 discusses what the evidence does and does not establish, and Section 6 summarizes the demonstrated conclusions and remaining validation needs.

2. Related Work and Research Gap

This study connects four closely related research streams: GA parameter setting in the PFSP, automated algorithm configuration, adaptive parameter control, and transfer or generalization of optimization knowledge. Their common premise is that algorithm performance depends not only on the search mechanism itself but also on how algorithmic choices are configured and how reliably knowledge obtained from previous problems can be reused. The streams differ, however, in the decision they seek to support: selecting parameter values before a run, adapting them during a run, learning configuration policies across tasks, or deciding whether previously acquired knowledge should be transferred to a new target.

2.1. GA Parameter Setting and Adaptive Control in Permutation Flow Shop Scheduling

GAs are a natural choice for the PFSP because a schedule can be represented directly as a job permutation and crossover and mutation can operate on feasible sequences. Reeves [8] showed early that a GA could be effective for flow-shop sequencing, and later work demonstrated that population management and genetic operators can materially affect solution quality [9]. Broad PFSP comparisons likewise show that implementation details and computational budgets matter alongside the high-level algorithm family [3].

More recent PFSP studies place genetic and population-based search within a broader and increasingly hybrid solution landscape. Zou et al. [18] combined clustering with a GA for multimodal PFSP optimization, while Lu et al. [19] combined deep reinforcement learning with an improved GA for a green PFSP variant. Dong et al. [20] likewise developed a deep-reinforcement-learning approach for minimizing late work in a flow shop. Other recent work has pursued decomposition, mathematical formulations, constructive heuristics, and iterated search [4,5]. Together, these studies show that PFSP solver design remains active and increasingly multi-component.

The importance of parameter setting within such solvers has a long history in evolutionary computation. Eiben et al. [6] distinguished parameter tuning, in which values are fixed before a run, from parameter control, in which values change during a run. Eiben and Smit [7] subsequently treated parameter tuning as an experimental and optimization problem in its own right, emphasizing parameter interactions and the limited generalizability of settings selected in a restricted problem region. This distinction remains relevant because a parameter vector that performs well during calibration need not retain the same quality when the problem environment changes.

Scheduling-specific studies provide a parallel illustration of this configuration-effort problem. Fığlalı et al. used design of experiments to study Ant System control parameters on job-shop scheduling problems and showed that structured experimental designs can reduce tuning effort while retaining competitive solution quality [21]. More recently, Sánchez Márquez et al. selected key hyperparameters of an evolutionary strategy for a mixed no-idle permutation flow-shop problem through literature-guided systematic experimentation and further reduced unnecessary computation through an experimentally derived stopping criterion without compromising solution quality [22]. Together, these studies reinforce that both parameter choice and the computational effort devoted to configuration remain practical concerns in scheduling metaheuristics. The present study addresses the subsequent transfer decision: once a configuration has been calibrated, how much additional tuning effort should be spent when that knowledge is transferred to a new problem size?

Recent research extends this idea from conventional offline tuning toward learned and adaptive parameter management. Zhou et al. [14] proposed a deep-reinforcement-learning-based parameter-control framework for multiobjective evolutionary computation, illustrating the growing use of learned mechanisms for adaptive parameter management. This line of work reflects a broader shift from fixed offline settings toward data-driven parameter adaptation during optimization.

A similar shift is now visible directly in PFSP research. Qi et al. [15] identified limited adaptability of statically configured control parameters as a weakness when the optimization landscape varies and introduced a Q-learning mechanism that adjusts key parameters dynamically during the search. Their study demonstrates that parameter adaptability remains an active concern in contemporary PFSP metaheuristics. Such online approaches address parameter sensitivity by modifying configurations during a run. The present study addresses a different but complementary question: whether a previously calibrated static GA configuration can be reused reliably on a new PFSP size before additional target-specific tuning is undertaken.

In scheduling studies, parameters are often tuned on a limited set of problems and the selected values are then used on a broader benchmark set. This is practical, but it can blur two different questions: 'Was this setting good on the calibration problems?' and 'Does this setting also remain within an acceptable loss tolerance on new problems?' The second question is the transferability question addressed here. Historical experiments in this study therefore measure not only good parameter centers but also how often transferred settings remain within the prespecified loss margin on unseen problems.

2.2. Automated Algorithm Configuration, Generalization, and Knowledge Transfer

Automated algorithm configuration (AAC) treats parameter setting directly as an optimization problem. Methods such as ParamILS and irace direct computational budget toward promising configurations rather than evaluating all candidates equally [10,11]. This is a major advantage over manual trial-and-error in large configuration spaces. A standard configurator, however, normally begins from the premise that tuning will be performed and then determines how the available tuning budget should be allocated. Racing and sequential model-based configuration provide related strategies for allocating configuration effort efficiently [12,13].

The broader automated-algorithm-design literature has increasingly emphasized not only configuration quality but also generalization. Ma et al. [16] organize the emerging Meta-Black-Box-Optimization literature around algorithm selection, algorithm configuration, solution manipulation, and algorithm generation, with algorithm configuration concerned with adapting hyperparameters and/or operators to the problem being solved. Their synthesis also highlights generalization to problems beyond the training distribution and computational efficiency as central considerations for learned optimization policies. This perspective reinforces the need to evaluate configuration knowledge on targets that were not used to construct it.

A related stream concerns algorithm selection and instance-specific configuration. Foundational algorithm-selection work formalized the use of problem characteristics to guide algorithm choice [23], while ISAC extended this logic to instance-specific algorithm configuration [24]. Subsequent meta-learning and automated-algorithm-selection research has shown how problem characteristics can support algorithm choice and increasingly instance-aware configuration decisions [25,26]. This literature recognizes that nominally similar problems can differ structurally and that problem characteristics may therefore carry useful information for configuration decisions.

Transfer optimization provides an additional perspective that is especially relevant to the present study. Xue et al. [17] define solution transferability in evolutionary optimization and show empirically that effective knowledge reuse depends on what is transferred, when transfer is performed, and how transfer is implemented. Their results emphasize that transferability should be assessed rather than inferred solely from source-task success, because indiscriminate transfer can create negative transfer while appropriate assessment can promote beneficial reuse. Although they focused on solution transfer rather than parameter configurations, the conceptual distinction is directly relevant here: strong performance on source problems does not by itself establish that the transferred knowledge will remain effective on a new target.

These developments suggest two distinct ways of responding to configuration uncertainty. One is to adapt or relearn parameters online, as in learned parameter-control approaches [14,15]. Another is to assess whether previously acquired knowledge is sufficiently reliable to reuse before committing additional computational effort. The latter is the focus of this paper. Instead of learning a continuously changing configuration policy, the

proposed framework asks whether a previously calibrated static configuration should be reused directly, locally refined, or replaced through full retuning.

Structural indicators derived from the processing-time matrix are also analyzed in this study, but they do not change the GREEN/AMBER/RED action. An instance-specific decision rule would require evidence that the selected features predict transfer behavior on genuinely unseen problems. The present framework therefore takes a conservative route: operational decisions are based on historically validated size anchors, while structural descriptors are retained in an explanatory role to help interpret flatter or sharper parameter-response surfaces.

2.3. Research Gap and Positioning of the Present Study

Existing literature provides mature methods for tuning algorithm parameters, allocating configuration budgets, adapting parameters online, and conditioning algorithmic decisions on problem characteristics [6,7,10–13,23–26]. Transfer-optimization studies further show that knowledge reuse should be assessed rather than assumed because source-task success does not guarantee beneficial transfer [17]. These streams establish the broader methodological foundations for the present work.

The remaining gap is narrower and operational. A practitioner may already have a calibrated static GA configuration from earlier PFSP studies but still face a new size for which a complete retuning exercise is expensive. Standard automated configurators primarily address how to search the configuration space once tuning is undertaken, while online parameter control addresses how settings change during a run. Neither question is identical to deciding, before target-specific tuning begins, whether existing parameter knowledge should be reused directly, locally checked, or discarded in favor of full retuning.

The framework studied here addresses that decision by combining two historical outputs that are deliberately kept separate. The parameter center is a reusable configuration recommendation. Historical transfer reliability is an empirical resampling index that summarizes how often transfer remained within the practical-loss margin on held-out same-size instances. For an off-grid target size, both are interpolated from surrounding anchors, but only the reliability index determines the amount of target-specific tuning.

This design is related to instance-specific configuration and algorithm selection in that previous evidence is used to support a new decision [23–26]. It differs in two respects. First, the operational input of the prespecified rule is problem size rather than a learned instance-level classifier. Second, the output is not an algorithm identity or a predicted best parameter vector; it is a tuning-effort action that determines risk-dependent search breadth and replication effort. Structural descriptors are retained only as diagnostic variables rather than action inputs.

The empirical contribution is likewise prospective rather than purely retrospective. The size-to-risk mapping, action thresholds, and conservative prospective validation procedure were fixed before the 24 off-grid outcomes were observed. Historical transfer reliability can therefore be tested against direct-center outcomes that were not used to construct it. The final staged screening-and-confirmation schedule is a computational refinement evaluated subsequently on the exactly recovered 24-target panel and is reported separately from the original confirmatory endpoints.

The study does not claim that the GREEN/AMBER/RED policy is a universally superior configurator. The original prespecified confirmatory comparison with direct center reuse does not establish statistical superiority. After exact recovery of the original 24 prospective matrices, the risk map was implemented with a lower-cost staged screening-and-confirmation schedule and compared with a reliability-blind tuner under an essentially identical total tuning budget. This later comparison supports the recom-

mended operational implementation but remains descriptive rather than a new prespecified confirmatory endpoint.

This positioning places the contribution between universal parameter reuse and universal retuning. The methodological object being validated is the decision process that links historical transfer evidence to configuration effort, not a new GA operator, a new online-control mechanism, or a claim that one parameter vector is optimal across PFSP instances.

3. Materials and Methods

The method has four steps. First, historical experiments provide a parameter center and transfer-reliability estimate for each of nine size anchors. Second, when a new problem size arrives, those quantities are interpolated from the surrounding anchors. Third, the estimated reliability is converted into a GREEN, AMBER, or RED action that determines search breadth. Fourth, the recommended operational implementation applies risk-dependent staged screening and finalist confirmation within that region. The original prospective experiment used a more conservative replication schedule to validate the risk rule; the lower-cost staged implementation is evaluated separately against a budget-matched reliability-blind tuner. This separates the questions 'Where should the search be concentrated?' and 'How much tuning evidence should be collected?'.

For readability, four terms should be separated from the outset. An 'anchor' is an $n \times m$ size that has already been studied in detail. The 'center' is the starting parameter setting transferred from that size to a new target. 'Reliability' is the historical rate at which this type of transfer remained within the prespecified practical-loss tolerance. The 'action' is the GREEN, AMBER, or RED decision that determines the breadth and replication depth of target-specific tuning.

Operationally, the decision module takes the target size (n, m), the prespecified nine-anchor table (parameter centers, historical reliability values, and generation budgets), and the prespecified GA parameter grid as inputs. It outputs the snapped starting center, interpolated reliability score, GREEN/AMBER/RED action, risk-dependent search region, staged replication schedule, generation budget, and final selected configuration. The portable reference implementation in Supplementary S1 uses a C++17 GA engine compiled with g++ = 13.3.0 and Python 3 orchestration. The distributed environment specification requires NumPy = 1.24, pandas = 2.0, and openpyxl = 3.1 for matrix generation, data handling, and spreadsheet I/O. No machine-learning library is used because the policy is a deterministic decision rule rather than a trained classifier. Historical production-environment metadata that were not retained are not reconstructed retrospectively; the supplied reference implementation supports deterministic replay and verification rather than historical wall-clock reproduction.

3.1. PFSP Model, GA Architecture, and Configuration Space

Let $p_j(k)$ denote the processing time of job j on machine k, and let $\pi = (\pi_1, \ldots, \pi_n)$ denote the common job sequence used on all machines. Let $C_{j,k}$ denote the completion time on machine k of the job occupying position j in π. The completion times and makespan are computed using the recursion in Equation (1):

$$C_{j,k} = \max\{C_{j-1,k}, C_{j,k-1}\} + p_{\pi j,k}, \qquad j = 1, \ldots, n;\ k = 1, \ldots, m, \qquad C_{max} = C_{n,m} \tag{1}$$

Boundary conditions are $C_{0,k} = 0$ for $k = 1, \ldots, m$ and $C_{j,0} = 0$ for $j = 1, \ldots, n$.

The GA architecture was fixed before the transfer study began. The purpose was to ensure that observed performance differences were due to parameter settings rather than changes in algorithmic components. Table 1 summarizes the fixed architecture and the parameter space considered.

Table 1. Fixed GA architecture and parameter space.

Component	Fixed Specification
Initialization	Random population
Selection	Binary tournament
Crossover	Fixed-cardinality OBX; k = $\lfloor n/2 \rfloor$ retained positions
Mutation	Insertion/shift mutation
Elitism	1
Replacement	Generational
Local search	None
Population ratio Ps/n	{1, 2, 3, 4, 5}
Crossover probability pc	{0.85, 0.8875, 0.925, 0.9625, 1.00}
Mutation probability pm	{0.025, 0.050, 0.075, 0.100, 0.125}
Full configuration grid	5 × 5 × 5 = 125 configurations
Main replications per configuration	R = 10 independent runs

The fixed kernel was deliberately kept simple and permutation-feasible. This isolates the effects of population size, crossover probability, and mutation probability from changes in the underlying algorithmic architecture.

Binary tournament provides a simple selection mechanism, while Order-Based Crossover (OBX) and insertion/shift mutation operate directly on permutation representations. Fixing the OBX cardinality at k = $\lfloor n/2 \rfloor$ also avoids introducing crossover strength as another tuning dimension.

Elitism of one preserves the incumbent best solution. Random initialization and the exclusion of embedded local search were used as methodological controls to avoid constructive-initialization and hybrid-search effects. The overall architecture is consistent with established permutation-based genetic search in the PFSP literature and with the broader principle of separating algorithm design from parameter-tuning effects [7–9].

Population size is expressed as the ratio Ps/n rather than as an absolute count so that settings remain comparable across job sizes. For example, Ps/n = 3 corresponds to a population of 150 on a 50-job problem and 300 on a 100-job problem. The same level therefore has a similar relative meaning across sizes.

The Nawaz–Enscore–Ham (NEH) heuristic of Nawaz et al. [27] is not used to initialize the GA or to select parameters. In this study, its makespan serves only as a deterministic normalization denominator, allowing performance differences to be expressed on a comparable scale across problem sizes. The fixed GA specification, reference implementation, deterministic seed rules, and reproducibility utilities are provided in Supplementary S1.

3.2. Historical Calibration and Construction of the Nine Size Anchors

Historical evidence is organized around nine Taillard sizes: n ∈ {20, 50, 100} and m ∈ {5, 10, 20}. Each size group contains ten benchmark instances [2]. On every instance, all 5 × 5 × 5 = 125 GA parameter combinations were evaluated, with 10 independent replications per combination. Two distinct quantities were fixed for each size anchor: the parameter center to be transferred to a new target and a group-level measure of how reliable this type of transfer had been historically. These are not the same quantity; the reliability estimate should not be interpreted as the success rate of one fixed center.

Generation depth G was calibrated separately for each anchor group before the 125-configuration response-surface experiment. Calibration compared increasing generation budgets using parameter-rank stability and the marginal performance gains obtained as G increased; proximity to a best-known PFSP solution was not used.

Once selected, G was fixed for that group before the 125-configuration main experiment and the historical transfer analysis. Thus, G is a size-specific search-budget control rather than a fourth tuned dimension of the 5 × 5 × 5 grid.

Because population size changes the number of evaluations performed per generation, the experiment is a parameter-policy study rather than an equal-evaluation comparison among configurations.

For each size, historical transfer reliability was estimated from all 120 possible three-instance calibration subsets of the ten benchmark instances. For each subset, the remaining seven instances were treated as unseen targets, yielding 120 × 7 = 840 resampling-based transfer comparisons per group. A comparison was counted as within tolerance when the transferred configuration's signed excess under independent validation did not exceed the prespecified 0.25-percentage-point (pp) margin relative to the target-specific empirical reference. The 840 comparisons reuse the same ten benchmark instances across overlapping calibration subsets and therefore are not statistically independent Bernoulli observations. Accordingly, the resulting rate is treated as an empirical resampling-based reliability index for the decision rule, not as a sample proportion from 840 independent trials or as a basis for binomial precision claims.

Representative instances were selected using five structural descriptors computed exclusively from the processing-time matrix. Together, they capture job-load heterogeneity, machine-load heterogeneity, bottleneck dominance, within-machine processing-time variability, and similarity between machine processing-time profiles. Table 2 summarizes their definitions.

Table 2. Structural descriptors used for representative-instance selection.

Code	Structural Interpretation	Computational Definition
CV_J	Job-load heterogeneity	$J_j = \Sigma_k\, p_{j,k}$ $CV_J = SD(J_1, \ldots, J_n)/Mean(J_1, \ldots, J_n)$
CV_M	Machine-load heterogeneity	$M_k = \Sigma_j\, p_{j,k}$ $CV_M = SD(M_1, \ldots, M_m)/Mean(M_1, \ldots, M_m)$
BI	Bottleneck dominance	$BI = max_k\, M_k/Mean(M_1, \ldots, M_m)$
CV_{WM}	Within-machine variability	$CV_{WM} = (1/m)\, \Sigma_k\, [SD_j\, p_{j,k}/Mean_j\, p_{j,k}]$
$Corr_M$	Inter-machine similarity	Mean Pearson $Corr(P_k, P_\ell)$ over all $k < \ell$

No GA output, NEH permutation, best-known solution, or other performance information entered these descriptors. Within each nominal n × m anchor group, the five descriptors were standardized before the Euclidean representativeness calculation.

The center for each size anchor was selected from three representative instances in the standardized five-dimensional structural space. The representative set minimized the largest distance from any instance to its nearest representative.

A robust-minimax rule was then applied to the three response surfaces. Configurations within 0.25 percentage points of the best mean loss were retained, and the candidate with the smallest worst loss across the three calibration instances was selected as the center.

Ties were resolved first by mean loss and then by ascending parameter order. The resulting nine centers and the group-specific generation budgets G were fixed before the prospective experiment began. The historical calibration and transfer-validation records underlying the nine-anchor reliability map, together with the corresponding quality-assurance material, are provided in Supplementary S2.

Table 3 therefore contains two distinct kinds of information: the parameter center transferred from each size and the historical near-tolerance transfer rate associated with that size. The final column, G, is the fixed generation budget. Risk-calibrated in this paper

means that empirical historical transfer evidence is mapped to prespecified tuning actions; it does not mean that the interpolated score is a statistically calibrated probability for an individual future instance. The canonical 100 × 20 historical input is 595/840 (70.83%), and the complete 840-comparison record yielding this count has been reconstructed and independently verified. A 597/840 result obtained under an alternate normalization is retained only as a sensitivity artifact and is not used for the reported anchor.

Table 3. Prespecified historical size anchors used by the transfer rule.

Anchor	Ps/n	pc	pm	Within-Tolerance Count	Within-Tolerance Rate	G
20 × 5	4	1.0000	0.125	749/840	89.17%	9000
20 × 10	5	0.8500	0.100	775/840	92.26%	8000
20 × 20	5	0.8875	0.075	720/840	85.71%	8000
50 × 5	2	1.0000	0.125	824/840	98.10%	2000
50 × 10	3	0.8500	0.100	656/840	78.10%	9000
50 × 20	5	0.9625	0.125	613/840	72.98%	9000
100 × 5	2	0.8500	0.075	837/840	99.64%	9000
100 × 10	3	0.8500	0.075	814/840	96.90%	9000
100 × 20	2	0.8875	0.050	595/840	70.83%	17,000

3.3. Size-Based Interpolation and the Recommended Risk-Guided Staged Tuning Policy

When a new problem has a size that was not directly studied historically, the four surrounding anchors are used. Let n_L and n_U denote the lower and upper neighboring job-count anchors and let m_L and m_U denote the corresponding lower and upper neighboring machine-count anchors. As a prespecified modeling choice, interpolation is performed in logarithmic coordinates to represent relative rather than absolute changes in problem size across the anchor grid. The target's relative position between the neighboring anchors is represented by the logarithmic coordinates u and v, defined in Equations (2) and (3), respectively:

$$u = [\ln(n) - \ln(n_L)]/[\ln(n_U) - \ln(n_L)] \quad (2)$$

$$v = [\ln(m) - \ln(m_L)]/[\ln(m_U) - \ln(m_L)] \quad (3)$$

For any anchor quantity q—historical reliability, Ps/n, pc, pm, or the generation budget G—the interpolated target value is denoted by $\hat{q}$ and obtained from the four neighboring anchor values using the bilinear interpolation in Equation (4):

$$\hat{q} = (1-u)(1-v)q_{LL} + (1-u)vq_{LU} + u(1-v)q_{UL} + uvq_{UU} \quad (4)$$

The interpolation has two distinct uses. For historical transfer reliability, the interpolated value is denoted by $\hat{s}$ and interpreted as an empirical historical within-tolerance rate for the target size under the prespecified 0.25-pp margin. For Ps/n, pc, pm, and G, interpolation provides the transferable starting point for the new target. The value $\hat{s}$ is therefore an operational decision score derived from historical resampling evidence rather than a formally calibrated probability of success. Importantly, using same-size historical reliability as a risk signal for an interpolated off-grid center is not assumed to be valid a priori; it is treated as an empirical modeling hypothesis and evaluated prospectively against direct-center loss on the 24 off-grid targets.

The three GA parameters are snapped to the nearest levels actually tested in the historical grid, avoiding an assumption that untested intermediate values behave smoothly. G is rounded to the nearest 1000 generations. The framework is validated only for interpolation inside n = 20–100 and m = 5–20; extrapolation beyond that envelope is not claimed.

The final step converts the interpolated historical reliability estimate $\hat{s}$ into a practitioner action. The 0.90 and 0.75 cutoffs were fixed before prospective outcomes were observed and increasing transfer-risk regions were defined rather than calibrated probabilities. In the final recommended implementation, those same regions control both search breadth and replication effort through a staged screening-and-confirmation procedure. The colors in Table 4 therefore describe how broadly and how deeply to tune, not the quality of the eventual solution.

Table 4. Recommended risk-guided staged tuning actions.

Action	Interpolated Historical Reliability, $\hat{s}$	Recommended Practitioner Action	Operational Tuning-Budget Cap
GREEN	$\hat{s} \geq 0.90$	Center + available ±1 axial neighbors; Stage-1 R = 5; confirm raw top 3 with reps 6–10	≤50 runs
AMBER	$0.75 \leq \hat{s} < 0.90$	±1 Cartesian neighborhood (≤27); Stage-1 R = 3; confirm raw top 8 with reps 4–10	≤150 runs
RED	$\hat{s} < 0.75$	Full 125-configuration grid; Stage-1 R = 2; confirm raw top 30 with reps 3–10	≤500 runs

Search breadth increases with estimated transfer risk. GREEN evaluates the transferred center and all available ±1 axial neighbors on the parameter grid; AMBER evaluates the ±1 Cartesian neighborhood around the center, containing at most 27 configurations; and RED retains full coverage of all 125 configurations. Boundary values are clipped to the available grid.

Replication depth is also staged. GREEN uses R = 5 for screening and sends the raw top 3 candidates to confirmation with new replications 6–10. AMBER uses R = 3 for screening and confirms the raw top 8 with new replications 4–10. RED uses R = 2 for screening and confirms the raw top 30 with new replications 3–10. Within each stage, candidates are ranked by mean C_{max} with ascending Ps/n, pc, and pm as deterministic tie-breakers. Final selection uses the lowest confirmation-only mean C_{max} rather than the cumulative screening-plus-confirmation mean, so the screening observations do not dominate the final comparison.

For practitioner-facing use, these requirements are summarized as operational tuning-budget caps of ≤50 runs for GREEN, ≤150 for AMBER, and ≤500 for RED. Under the prescribed screening-and-confirmation schedule, the corresponding exact maximum run counts are 50, 137, and 490, respectively. Historical transfer evidence supplies the risk ordering and motivates increasing search breadth; the lower-replication screening-and-confirmation schedule is the operational refinement evaluated on the 24-target panel in Section 3.5. The original prospective experiment used the more conservative validation implementation described in Section 3.4 and is retained unchanged for confirmatory inference. Algorithm 1 consolidates the preceding interpolation, risk-classification, and staged screening-and-confirmation rules into the complete practitioner workflow for a new PFSP size.

Consider, for illustration, a 33 × 12 PFSP. Because this target lies between the 20- and 50-job anchors and the 10- and 20-machine anchors, the four surrounding historical anchors are 20 × 10, 20 × 20, 50 × 10, and 50 × 20. Their historical within-tolerance rates are 92.26%, 85.71%, 78.10%, and 72.98%, respectively (Table 3). Applying the logarithmic coordinates in Equations (2) and (3) and the bilinear interpolation in Equation (4) to these four values yields an estimated historical reliability of 83.00%. Since this value lies between the prespecified thresholds of 0.75 and 0.90, the target is assigned to the AMBER region. The same interpolation procedure is applied to Ps/n, pc, pm, and G. After the interpolated parameter values are snapped to the nearest levels of the prespecified grid, the resulting

starting center is (Ps/n, pc, pm) = (4, 0.8875, 0.10), corresponding to Ps = 132 for n = 33, while the generation budget is rounded to G = 9000. Under the recommended AMBER action, the practitioner screens the local Cartesian neighborhood at R = 3 and confirms the raw top 8 with new replications 4–10, requiring at most 137 tuning runs under the specified schedule, which falls within the AMBER operational cap of ≤150 runs, rather than the 1250 runs needed for uniform full-grid retuning. This example is provided solely to illustrate the application of the decision rule and is not part of the 24-target prospective validation panel.

Algorithm 1. Recommended practitioner procedure for a new PFSP size

Input: target size (n, m) inside the validated anchor envelope; prespecified size anchors and GA parameter grid.
1. Locate the lower and upper neighboring anchors for n and m in logarithmic size coordinates.
2. Compute the relative coordinates u and v and bilinearly interpolate historical transfer reliability $\hat{s}$, Ps/n, pc, pm, and the generation budget G.
3. Snap Ps/n, pc, and pm to the nearest prespecified grid levels and round G to the nearest 1000 generations.
4. If $\hat{s} \geq 0.90$, assign GREEN; screen the center and available ±1 axial neighbors at R = 5; confirm the raw top 3 with reps 6–10.
5. Else if $0.75 \leq \hat{s} < 0.90$, assign AMBER; screen the ±1 Cartesian neighborhood at R = 3; confirm the raw top 8 with reps 4–10.
6. Else assign RED; screen all 125 configurations at R = 2; confirm the raw top 30 with reps 3–10.
7. In every action, select the finalist with the lowest confirmation-only mean C_{max}, using ascending parameter values as deterministic tie-breakers.
8. Run the final GA with the selected configuration and transferred generation budget.
Output: selected GA configuration, tuning action, and generation budget for the target size.

3.4. Prospective Off-Grid Validation and Statistical Analysis

The prospective panel contains 24 off-grid size combinations formed by n ∈ {30, 40, 60, 70, 80, 90} and m ∈ {5, 10, 15, 20}, with one target matrix per n × m combination. This prespecified design deliberately favored coverage of 24 distinct off-grid sizes over repeated matrices within fewer sizes under the available experimental budget; it therefore tests size-space transfer more broadly but cannot estimate matrix-to-matrix variation within a fixed size. The protocol specified independent and identically distributed (iid) integer processing times on {1, ..., 99}, NumPy PCG64DXSM, and matrix seed 2026082701 + n × 1000 + m, with the first generated matrix used without screening. A provenance audit identified an omitted implementation detail in the integer draw path: the original matrices were generated through a 16-bit integer stream rather than the default 64-bit integer path. Regeneration with the prespecified seeds and the recovered 16-bit draw detail reproduced all 24 archived NEH values exactly. This recovery closes the matrix-provenance gap without changing the prospective target panel, action assignments, or any originally reported outcome; the recovery audit and regenerated matrices are provided in Supplementary S1 and S2.

The original prospective validation retained the conservative implementation fixed before target outcomes were observed: GREEN used the transferred center without target-specific tuning; AMBER screened its local neighborhood at R = 3 and extended the raw top 5 to R = 10, for at most 116 runs; RED evaluated all 125 configurations at R = 10, for 1250 runs. This conservative schedule remains the basis of the confirmatory 21/24 policy-versus-center analysis reported below. It is preserved as validation evidence and should not be confused with the lower-cost staged implementation now recommended for operational use in Table 4 and evaluated in Section 3.5.

Validation of the original prospective policy used three independent stochastic data streams, and every decision or comparison was based on repeated runs rather than a single seed. The first stream was used only for AMBER or RED tuning under the con-

servative prospective implementation and therefore determined which configuration that policy selected.

After that selection is locked, a second, hidden stream evaluates the complete 125-configuration grid with R = 10 replications per configuration. The configuration with the smallest 10-replication mean C_{max} becomes the target-specific empirical reference; the policy does not see this hidden surface during selection. Within each target, all 125 configurations are evaluated under the same fixed target-specific generation budget G and identical stopping rule; across target sizes, G changes only through the prespecified interpolation and rounding rule described in Section 3.3.

A third stream is reserved for final validation. The policy-selected configuration, the target-specific empirical reference, and, when required, the direct center are rerun on the same third seed family with R = 10 under common random numbers. Selection, reference construction, and final assessment therefore use non-overlapping stochastic evidence.

Final performance is measured by signed normalized excess, abbreviated SE_{pp}. Let $\overline{C}_{policy}$ denote the 10-replication mean C_{max} of the policy-selected configuration on the final validation seeds, $\overline{C}_{ref}$ the corresponding 10-replication mean for the target-specific empirical reference on the same seeds, and C_{NEH} the NEH makespan for the same target. The overbar denotes an arithmetic mean across the 10 final-validation replications, and SE_{pp} is calculated as shown in Equation (5):

$$SE_{pp} = 100 \times (\overline{C}_{policy} - \overline{C}_{ref})/C_{NEH} \quad (5)$$

SE_{pp} expresses the policy-reference difference in percentage points on the target-specific NEH-normalized performance scale. Thus, $SE_{pp} = 0$ means equal final-validation means, $SE_{pp} = 0.25$ means that the policy is worse by an amount equal to 0.25% of the target's NEH makespan, and $SE_{pp} < 0$ means that the policy performed better on the independent validation seeds.

Negative values are possible because the empirical reference is selected from one stochastic data stream and both configurations are then rerun on another; their sample ordering can therefore reverse. The primary prespecified practical-tolerance criterion is $SE_{pp} \leq 0.25$ pp, labeled SAFE (within-tolerance). For example, when NEH $C_{max} = 3000$, 0.25 pp corresponds to about 7.5 C_{max} units. Clipped loss, $\max(0, SE_{pp})$, summarizes only positive excess.

SAFE is an operational classification of the observed fresh-seed mean difference, not a formal statistical equivalence test. The 0.25-pp threshold was fixed prospectively to provide a transparent decision tolerance; because the study does not impose an application-specific economic loss function, sensitivity is also reported at 0.10 and 0.50 pp.

Two classifications must not be confused. GREEN/AMBER/RED is the tuning action assigned before the target outcome is observed. The performance result observed after independent validation is classified according to whether it falls within the prespecified tolerance: $SE_{pp} \leq 0.25$ is called SAFE (within tolerance), whereas $SE_{pp} > 0.25$ is classified as above tolerance. A GREEN target can be above tolerance and a RED target can be SAFE; color is the action, whereas within/above tolerance is the observed performance outcome.

Two prespecified co-primary tests compare the policy with direct use of the interpolated center. A one-sided paired Wilcoxon signed-rank test evaluates whether target-level clipped loss is lower under the policy. A one-sided exact discordant-pair McNemar/binomial test evaluates whether the policy reduces the number of above-tolerance outcomes relative to direct center transfer. Both tests use $\alpha = 0.05$, with Holm adjustment across the two comparisons.

Overall policy performance is also summarized by the within-tolerance count and rate, Wilson 95% confidence intervals, mean signed excess, and mean clipped loss. Because

the GREEN, AMBER, and RED subgroups are small, subgroup results are interpreted mainly descriptively.

For the original prospective validation implementation, tuning effort is reported as the number of configuration runs: GREEN required zero target-specific tuning runs, AMBER used its realized local-tuning count, and RED required 1250 runs. Universal full retuning of all 24 targets would require $24 \times 1250 = 30{,}000$ tuning runs. These historical validation counts are retained to document the prespecified confirmatory experiment; the recommended staged operational policy has lower action-specific budgets and is evaluated separately in Section 3.5. Run counts are not CPU-time or wall-clock measures because generation depth and population size differ across targets and configurations. Hidden benchmark surfaces and final validation runs are excluded because they are needed only for scientific evaluation, not practical deployment.

The five structural descriptors and a predicted response-surface sharpness score were prespecified secondary diagnostics. They could not alter target generation, center interpolation, action assignment, thresholds, or prospective endpoints. Supplementary S1 provides the computational specifications, portable reference implementation, deterministic GA and matrix seed rules, exact 24-matrix recovery utilities, the original prospective protocol, and the staged operational-policy protocol; Supplementary S2 provides the retained historical transfer records, the complete canonical 100 × 20 problems, 840-comparison reconstruction, complete 24-target prospective results, the staged policy's budget-matched validation data, statistics, and quality-assurance (QA) material.

3.5. Operational Staged Policy and Budget-Matched Validation Against Reliability-Blind Tuning

The final recommended operational policy combines the prospectively validated risk map and risk-dependent search breadth with the lower-replication staged screening-and-confirmation schedule in Table 4. Its empirical evaluation uses the same 24 exactly recovered target matrices. To isolate the value of historical risk information from the value of tuning itself, the staged policy is compared with a reliability-blind tuner under an essentially identical total tuning budget. This analysis does not replace or retroactively modify the original prospective confirmatory endpoints; it evaluates the computationally more economical implementation recommended for future use.

For the operational staged policy, the transferred center is neither forced into the finalist set nor used as an acceptance gate. GREEN evaluates the transferred center and available ± 1 axial neighbors with five Stage-1 replications and confirms the raw top three using new replications 6–10; AMBER evaluates the ± 1 Cartesian neighborhood around the center with three Stage-1 replications and confirms the raw top eight using new replications 4–10; RED evaluates the full 125-configuration grid with two Stage-1 replications and confirms the raw top 30 using new replications 3–10. The method returns the finalist with the lowest confirmation-only mean C_{max}.

The reliability-blind comparator receives the same search logic at every target, irrespective of historical reliability. It evaluates all 125 grid configurations once, confirms the raw top eight with four additional replications, and selects the finalist with the lowest confirmation-only mean $C_{max.}$ This requires 157 tuning runs per target, or 3768 across 24 targets. Owing to boundary-dependent neighborhood sizes, the risk-guided total is 3767 runs, a one-run difference of 0.027% relative to the blind budget.

Guided and blind search use separate common-random-number seed families during tuning. After both configurations are locked, they are evaluated on the same 20 fresh common-random-number validation seeds per target; these validation observations are never used for search or selection. The primary descriptive contrast is the difference in fresh-validation mean C_{max}, with positive guided advantage defined as

$100 \times (\overline{C}_{\text{blind}} - \overline{C}_{guided})/\overline{C}_{\text{blind}}$. A ±0.05% practical-equivalence band is used only to aid interpretation of small differences and is not a tuning, selection, or hypothesis-testing threshold.

The transferred center is not forcibly included among finalists in either method, because mandatory inclusion would change the scientific question from allocation efficiency to protected center replacement. The same rule is applied to all 24 targets. The staged implementation is therefore recommended on the basis of its lower configuration-run burden and its observed matched-budget performance, while its guided-versus-blind comparison remains descriptive rather than a prespecified confirmatory test.

The evidential roles of the analyses are kept separate. The two all-panel comparisons specified before the prospective outcomes were observed remain the confirmatory tests of the original conservative implementation. Action-subgroup summaries, tolerance-sensitivity checks, finite-grid random baselines, historical deletion analyses, and the budget-matched evaluation of the staged operational policy are descriptive or robustness analyses because they were not specified as additional confirmatory endpoints. Here, descriptive means that observed effect magnitudes and patterns are reported without using them to claim population-level statistical superiority; it does not mean that the quantities are subjective or anecdotal.

4. Results

The results are organized around five questions. First, how variable are historically selected parameter settings, and how stable is the transfer-reliability evidence that underlies the rule? Second, does the interpolated reliability index rank prospective risk before tuning is applied? Third, what changes when the original conservative prospective policy invokes AMBER or RED tuning compared with direct center reuse? Fourth, how sensitive are the conclusions to action thresholds, performance tolerances, response-surface flatness, and stochastic selection uncertainty? Fifth, does the recommended staged operational implementation use the risk information effectively under a budget matched to reliability-blind tuning?

4.1. Historical Parameter-Landscape Heterogeneity and Transferability

The historical response surfaces do not support a single universal GA parameter vector. Within every one of the nine Taillard size groups, the target-specific empirical reference configuration varies across the ten instances, and the number of distinct reference configurations ranges from six to ten. This heterogeneity motivates evaluating transfer by performance loss rather than by exact recovery of the same sample-winning configuration. Table 5 summarizes the resulting heterogeneity of the historical empirical-reference configurations together with the corresponding fresh-seed transfer-reliability rates across the nine anchor groups.

Historical transfer can nevertheless remain reliable despite this pointwise heterogeneity. Across the nine size anchors, 6583 of 7560 resampling-based held-out comparisons remained within the 0.25-pp tolerance. The rates vary markedly by region, from 70.83% at 100 × 20 to 99.64% at 100 × 5, which is the empirical variation used by the subsequent effort-allocation rule.

The relevant distinction is therefore between identifying the lowest observed finite-grid mean on one stochastic response surface and identifying a region that continues to incur small loss after transfer. The target-specific empirical reference is generated by a deterministic rule—the grid configuration with the lowest ten-replication hidden mean C_{max}—and is therefore not a subjective judgment. It remains a finite-grid, finite-replication stochastic reference rather than a true optimum. A transferred configuration can differ from that reference configuration while still remaining within the prespecified practical-loss margin.

Table 5. Historical empirical-reference heterogeneity and fresh-seed transferability by anchor group.

Group	Modal Ps/n	Modal pc	Modal pm	Distinct References	Historical SAFE Rate
20 × 5	5	0.9625	0.125	10	749/840 (89.17%)
20 × 10	5	0.9625	0.100	9	775/840 (92.26%)
20 × 20	5	0.8875	0.125	8	720/840 (85.71%)
50 × 5	4	0.8500	0.075	10	824/840 (98.10%)
50 × 10	4	0.9250	0.125	9	656/840 (78.10%)
50 × 20	5	0.9625	0.100	8	613/840 (72.98%)
100 × 5	5	0.8500	0.125	10	837/840 (99.64%)
100 × 10	1	0.8500	0.125	8	814/840 (96.90%)
100 × 20	2	0.8500	0.075	6	595/840 (70.83%)

Historical sensitivity analyses show that the reliability map is informative but not perfectly stable. Strict one-instance deletion analyses were completed for all nine anchors, with 504 comparisons per deletion fold. Four anchors (20 × 5, 20 × 20, 50 × 20, and 100 × 20) cross an action threshold under at least one deletion. The exact 100 × 20 folds range from 66.67% to 79.17%, so its nominal RED anchor can enter AMBER when instance 10 is omitted. Propagating all nine anchors through the prespecified interpolation yields 90 single-anchor deletion scenarios, 19 action-flip events affecting eight targets, and 16/24 targets whose action is stable in every scenario. Because the reconstructed aggregate remains exactly 595/840, the prespecified nominal map itself is unchanged at 8 GREEN, 12 AMBER, and 4 RED targets.

4.2. Prospective Risk Ordering and Policy Performance

Before examining the effect of tuning, the historical reliability index was compared with the prospective outcome of direct center reuse, because direct-center performance is observed before any reliability-triggered intervention. Mean direct-center clipped loss increased monotonically from 0.0474 pp in GREEN to 0.0971 pp in AMBER and 0.3016 pp in RED. Across all 24 targets, reliability was negatively associated with direct-center clipped loss (Spearman $\rho = -0.521$, nominal two-sided $p = 0.009$). Across all 90 single-anchor deletion scenarios, this association remained negative (Spearman $\rho = -0.570$ to -0.478; median -0.521; nominal two-sided $p = 0.0037$–0.0183). This additional diagnostic supports the index as a risk-ranking surrogate within the tested envelope; it is not a probability-calibration test. Here, $V_{0.25}$ denotes the proportion of the 125 tested grid configurations whose loss lies within the 0.25-pp practical-tolerance margin. Table 6 summarizes the original prospective performance of the three risk classes under the conservative prespecified implementation, including within-tolerance rates, clipped loss, response-surface coverage, and tuning-run expenditure.

Table 6. Original prospective performance and tuning effort by risk class.

Action	Within Tolerance/N	Within Tolerance Rate	Wilson 95% CI	Mean Clipped Loss (pp)	Mean $V_{0.25}$	Tuning Runs
GREEN	8/8	100.0%	67.6–100.0%	0.0474	0.754	0
AMBER	10/12	83.3%	55.2–95.3%	0.0772	0.143	1185
RED	3/4	75.0%	30.1–95.4%	0.0830	0.134	5000
Overall	21/24	87.5%	69.0–95.7%	0.0682	—	6185

The prespecified risk rule classified eight targets as GREEN, 12 as AMBER, and 4 as RED. Under the conservative prospective validation implementation, the policy was within the prespecified 0.25-pp tolerance on 21 of 24 targets. All eight GREEN targets were within tolerance without target-specific tuning, but this zero-tuning result is partly explained by broad response surfaces: five of the eight hidden GREEN surfaces had $V_{0.25} = 1.000$, meaning every tested grid configuration lay within the tolerance.

Surface flatness does not explain the complete GREEN result. Across all eight GREEN hidden surfaces, the mean within-tolerance grid share was 0.754, so one independent uniform grid draw per target would be expected to produce 6.032 within-tolerance outcomes out of eight. The exact probability that eight independent uniform grid draws will all fall within tolerance is 0.6083%. The two sharpest GREEN targets, 30 × 5 and 80 × 10, had only 10/125 and 11/125 hidden configurations within the 0.25-pp margin, respectively, while the transferred centers remained within tolerance under independent final validation; the probability that independent uniform draws would be within tolerance on both of these sharp surfaces is 0.704%. These are exact finite-grid chance benchmarks conditional on the fully enumerated hidden surfaces, not confirmatory randomized-control *p*-values.

The original conservative prospective implementation required 6185 configuration runs: none for GREEN, 1185 across AMBER, and 5000 across RED. Universal 125-configuration retuning with ten replications on every target would require 30,000 runs, so that validation schedule reduced run count by 79.38%. Because a run at a larger population, generation depth, n, or m is more expensive than a run at a smaller setting, this percentage is not interpreted as an equal CPU-time reduction. Normalized workload proxies for the original schedule indicate reductions of about 66.7–67.3%, and a rough n × m-weighted proxy indicates about 51.1–51.6%. These values document the prospective validation protocol rather than the final operational budget; the recommended staged policy is evaluated in Section 4.5.

4.3. Sensitivity Analyses and Comparison with Direct Center Reuse

Two kinds of thresholds are relevant and are kept separate. The GREEN/AMBER/RED cutoffs determine risk class and search breadth, whereas the 0.25-pp practical-tolerance margin classifies final observed loss. Sensitivity analysis of the action cutoffs showed a nontrivial stability plateau around the prespecified rule: all 24 action labels remain unchanged for a GREEN threshold in $(0.8723, 0.9085)$ and a RED threshold in $(0.7241, 0.7608)$. Under the original conservative prospective cost schedule, a joint ± 0.025 neighborhood of the 0.90/0.75 pair changes at most three actions and implies 6185–8587 tuning runs. The final staged schedule retains the same ordinal risk map while corresponding to operational budget caps of $\leq$50, $\leq$150, and $\leq$500 runs for GREEN, AMBER, and RED, respectively.

Sensitivity to the final performance tolerance gives a different view. As summarized in Table 7, the policy is within tolerance on 19/24 targets at 0.10 pp, 21/24 at 0.25 pp, and 23/24 at 0.50 pp; direct center reuse is within tolerance on 13/24, 21/24, and 23/24, respectively. Thus, the 0.25-pp equality in counts is a coarse binary summary rather than evidence that the two strategies are equivalent.

Table 7. Sensitivity of prospective performance classification to the practical-tolerance margin.

Practical-Tolerance Margin	Policy Within Tolerance	Policy Rate	Direct-Center Within Tolerance	Direct-Center Rate	Policy Targets Above Tolerance
0.10 pp	19/24	79.2%	13/24	54.2%	30 × 5; 60 × 15; 70 × 15; 80 × 10; 90 × 20
0.25 pp	21/24	87.5%	21/24	87.5%	60 × 15; 70 × 15; 90 × 20
0.50 pp	23/24	95.8%	23/24	95.8%	70 × 15

Direct center reuse is the principal comparator because it requires no target-specific tuning and uses exactly the same transferred center from which the policy starts. At the primary 0.25-pp margin, both strategies are within tolerance on 21/24 targets, but the target-level pattern behind this tie is more informative than the aggregate count alone. Table 8 identifies the discordant and persistent above-tolerance cases. The discordant targets are 70 × 20, where RED tuning recovers an above-tolerance center, and 90 × 20, where the center is just inside the margin while the RED-selected configuration is just outside.

Table 8. Diagnostic comparison of discordant and above-tolerance targets.

Size	Action	Policy Result	Center Result	Interpretation
60 × 15	AMBER	0.3495; Above	0.3272; Above	Both miss the 0.25-pp margin
70 × 15	AMBER	0.5439; Above	0.3164; Above	Only policy result above tolerance at 0.50 pp
70 × 20	RED	−0.0881; Within	0.7409; Above	Full tuning brings the center result within tolerance
90 × 20	RED	0.2611; Above	0.2396; Within	Reverse discordance; marginal crossing

The recovery at 70 × 20 is substantial: the direct center has signed excess 0.7409 pp, whereas the RED-selected configuration has −0.0881 pp on the independent final-validation stream. The reverse discordance at 90 × 20 is much smaller, changing from 0.2396 pp for the center to 0.2611 pp for the policy, only 0.0111 pp beyond the primary tolerance. The 60 × 15 and 70 × 15 AMBER targets remain above tolerance under both strategies.

The incremental comparison is most informative on the 16 AMBER+RED targets, because those are the targets on which the policy actually spends tuning effort. Mean clipped loss is 0.0786 pp for the policy versus 0.1483 pp for direct center reuse, a descriptive reduction of 46.95%; mean signed excess is −0.0236 pp versus 0.0740 pp. For clipped loss, 7 targets improve, 3 worsen, and 6 are unchanged; for signed excess, 10 improve, 5 worsen, and 1 is unchanged.

These intervention-only differences remain exploratory. The one-sided Wilcoxon p-value is 0.0571 for clipped loss and 0.0778 for signed excess. On the full 24-target panel, the prespecified clipped-loss Wilcoxon test has raw $p = 0.0571$ and Holm-adjusted $p = 0.1141$, while the exact discordant-pair test for the 0.25-pp classification has raw and adjusted $p = 0.75$. Statistical superiority over direct center reuse is therefore not established, and non-significance is not interpreted as evidence of equivalence.

At the stricter 0.10-pp margin, the contrast is more pronounced on the 16 intervention targets: 13/16 policy results are within tolerance compared with 7/16 direct centers, with six center-fail/policy-within recoveries and no reverse discordance. At 0.25 and 0.50 pp,

both approaches are within tolerance on 13/16 and 15/16 intervention targets, respectively. Taken together with the continuous-loss results, these findings show why the 21/24 equality at the primary margin should be interpreted as a coarse endpoint rather than as evidence of equivalent behavior. These tolerance analyses are additional robustness evidence rather than additional confirmatory endpoints.

4.4. Structural Diagnostics, Selection Uncertainty, and Boundary Cases

The structural descriptors were not allowed to alter the prespecified action, but they remain informative diagnostics. Predicted response-surface sharpness was strongly associated with realized hidden-surface sharpness across the 24 targets (Pearson $r = 0.872$; Spearman $\rho = 0.904$; both nominal $p < 0.001$). This relationship motivates future structure-aware policies, but it does not validate the descriptors as current action variables or prove that they predict transfer failure.

The persistent above-tolerance AMBER cases, 60 × 15 and 70 × 15, show that limited local tuning is not guaranteed to recover a weak center. Their center results are already above the primary margin, and the local AMBER procedure does not move them inside it. These cases are therefore evidence against interpreting the action labels as deterministic quality guarantees.

The RED cases illustrate the opposite outcomes possible under full tuning. At 70 × 20, the full grid clearly recovers a poor center. At 90 × 20, the selected full-grid configuration is marginally worse on fresh seeds than the direct center. Because 125 stochastic candidates are compared using finite replications, the configuration with the smallest tuning-stream mean need not remain the smallest on an independent validation stream.

This same mechanism explains why negative signed excess is possible relative to the target-specific empirical reference. The reference is selected from the hidden stream and then rerun on the final stream; fresh-sample ordering can reverse. Accordingly, the empirical reference is a finite-grid evaluation benchmark rather than a noise-free optimum or deterministic lower bound.

Taken together, the original prospective results support a bounded claim. Historical reliability carries useful information about the risk of direct reuse, and additional tuning can reduce loss on some higher-risk targets. Those results validate the risk ordering and the increasing search-breadth logic, but they do not require retaining the original conservative replication counts as the operational recommendation. Section 4.5 therefore evaluates the lower-cost staged implementation in Table 4 against a reliability-blind tuner under an essentially identical total budget.

4.5. Budget-Matched Validation of the Recommended Staged Risk-Guided Policy

The budget-matched experiment evaluates the recommended staged policy while holding total tuning effort essentially constant against a reliability-blind comparator. The risk-guided staged policy used 3767 tuning runs and the reliability-blind strategy used 3768. Relative to universal 125 × 10 retuning, the staged policy therefore reduces configuration-run count by 87.44%; it also uses 39.09% fewer runs than the 6185-run conservative prospective validation schedule. Table 9 summarizes both the allocation of those runs and the fresh-validation outcomes, with G/Tie/B denoting guided-better, practically tied, and blind-better outcomes, respectively. Because both selected configurations were evaluated on the same 20 new seeds for every target, the comparison measures out-of-sample performance after method-specific search rather than reuse of tuning observations.

Across the 24 targets, the risk-guided configuration had the lower fresh-validation mean C_{max} on 16 targets, the blind configuration on 5, and the means were exactly tied on 3. Using an absolute guided relative advantage of $\leq$0.05% as an interpretive practical-

equivalence band, the corresponding classification was 9 guided-better, 14 practically tied, and 1 blind-better. Positive guided relative advantage denotes a lower fresh-validation mean C_{max} under guided allocation. The ±0.05% band was used only for interpretation and played no role in configuration selection. The mean guided relative advantage across targets was 0.0503%, and the median was 0.0372%.

Table 9. Budget-matched comparison of risk-guided and reliability-blind tuning.

Risk Group	N	Guided Runs	Blind Runs	G/Tie/B	Guided Advantage Mean/Median (%)
All	24	3767	3768	9/14/1	0.0503/0.0372
GREEN	8	370	1256	3/4/1	0.0518/0.0212
AMBER	12	1437	1884	4/8/0	0.0517/0.0322
RED	4	1960	628	2/2/0	0.0429/0.0530

The matched total budget is achieved by reallocating search breadth and replication effort rather than by increasing total cost. Relative to the blind schedule, the staged guided policy uses markedly fewer runs on GREEN targets (370 versus 1256) and fewer on AMBER targets (1437 versus 1884), while concentrating more effort on RED targets (1960 versus 628). Under the practical-equivalence interpretation, guided allocation is better on 3/8 GREEN, 4/12 AMBER, and 2/4 RED targets; blind allocation is better on only one GREEN target. These results support the staged schedule as the more economical operational implementation within the tested panel, while the guided-versus-blind comparison remains descriptive rather than a new confirmatory endpoint.

5. Discussion

The central finding is not that one tuning strategy dominates all alternatives. Instead, the evidence supports treating GA configuration as a two-part decision: first identify a reasonable parameter center, then decide how much additional evidence is worth collecting before that center is trusted. Historical transfer reliability contributes to the second decision by ranking reuse risk across the tested size space.

5.1. Historical Reliability as a Tuning-Effort Signal

Historical reliability and the transferred center play different roles. The center is a configuration recommendation derived from representative historical response surfaces; the reliability index is a resampling-based summary of how often historically selected configurations remained within the practical-loss margin on held-out same-size instances. The off-grid experiment tests whether that same-size evidence remains informative after interpolation rather than assuming that it does.

The prospective direct-center results provide the strongest support for this interpretation. Mean clipped loss rises from GREEN to AMBER to RED, and the continuous reliability score is negatively associated with direct-center clipped loss. This negative association also persists across all 90 single-anchor deletion scenarios ($\rho = -0.570$ to -0.478), strengthening the interpretation that the signal is not an artifact of one nominal anchor realization. The completed historical deletion analyses nevertheless show that the signal is not perfectly stable: anchors close to the 0.75 and 0.90 boundaries can change action under one-instance deletion. Reliability should therefore be read as an empirical risk-ranking surrogate, not as a calibrated future success probability.

The interpolation step is similarly pragmatic rather than a structural law. Log-bilinear interpolation was selected because it is deterministic, uses only the four surrounding an-

chors, reproduces anchor values exactly, and represents relative changes in n and m without fitting additional smoothing parameters. In sensitivity checks, raw-coordinate bilinear interpolation changes target reliability by 0.852 percentage points on average (maximum 1.971 pp), preserves 23/24 action labels, and yields a similar reliability-loss rank association (Spearman $\rho = -0.512$). Local inverse-distance weighting in log-size space ($p = 2$) changes reliability by 0.839 pp on average (maximum 2.822 pp), also preserves 23/24 labels, and yields $\rho = -0.526$. Historical leave-center-out checks nevertheless show substantial local residuals. The method is therefore best described as a transparent local smoother inside the anchor envelope rather than a structural law.

Action-threshold sensitivity reinforces this cautious interpretation. The prespecified 0.90/0.75 rule lies inside a nontrivial no-change rectangle, so modest threshold perturbations do not alter any of the 24 risk labels. Under the original conservative validation implementation, crossing the RED boundary created a large cost jump because full 125 × 10 retuning replaced at most 116 AMBER runs. The recommended staged policy preserves increasing search breadth with risk but smooths that computational escalation: the operational action-budget caps are ≤50 runs for GREEN, ≤150 for AMBER, and ≤500 for RED. The thresholds remain operational cutoffs for effort allocation rather than statistically estimated breakpoints.

5.2. What Additional Tuning Adds Beyond Direct Reuse

The primary 0.25-pp count alone does not show an advantage for the policy: both the policy and direct center reuse are within tolerance on 21/24 targets. That equality should not be interpreted as equivalence. The binary endpoint is relatively coarse, and the two strategies fail on different targets. Continuous loss and stricter-tolerance analyses provide additional information about what the tuning actions actually change.

On the 16 AMBER+RED targets where the policy intervenes, mean clipped loss is descriptively lower under the policy, and the 0.10-pp classification yields six recoveries with no reverse discordance. The 70 × 20 case shows the intended use of RED tuning, because a clearly above-tolerance center is recovered. The 90 × 20 case shows the opposite possibility: full tuning can select a configuration that loses a small apparent advantage on fresh seeds. The evidence therefore supports selective tuning as a risk-management action, not as a guarantee of target-wise improvement.

The GREEN analysis also clarifies why the zero-tuning success observed in the original prospective validation should not be overinterpreted. Five of the eight GREEN hidden surfaces are so broad that every grid configuration lies within the 0.25-pp margin, so surface flatness genuinely explains much of the 8/8 result. Across all eight GREEN surfaces, one uniform random grid draw per target would be expected to be within tolerance on 6.032 targets, and the exact probability that all eight independent random draws are within tolerance is 0.608%. At the same time, the two sharp GREEN cases contain only 8.0% and 8.8% within-tolerance grid configurations, yet the transferred centers remain within tolerance on independent validation. These diagnostics motivate the final operational choice to spend only a small verification budget in GREEN regions rather than either assuming that no tuning is ever needed or applying broad retuning.

This distinction matters for practitioners. The final policy does not equate GREEN with zero effort or RED with uniform ten-replication evaluation of every configuration. Instead, it assigns a small verification budget to GREEN, a moderate local-search budget to AMBER, and broader but still staged full-grid screening to RED. A practitioner with an explicit cost of computation and cost of schedule degradation could replace the current operational thresholds or replication schedule after separate calibration, but the present

recommendation is the staged design summarized operationally by ≤50/≤150/≤500 run caps; the corresponding exact schedule-derived maxima are 50/137/490 runs.

5.3. Computational Cost, Budget-Matched Allocation, and Stochastic Selection

The original prospective validation already reduced configuration work from 30,000 universal full-grid tuning runs to 6185 runs under its conservative schedule. That result remains useful as validation evidence, and its workload proxies indicate smaller savings than the raw run count once population size, generation depth, and problem dimensions are considered. The operational recommendation is now the staged policy, which further lowers the observed 24-target tuning requirement to 3767 runs. Run count remains a transparent and reproducible configuration-work measure rather than a direct estimate of wall-clock time.

The budget-matched validation became feasible after exact recovery of the original 24 prospective matrices. The recommended staged policy used 3767 tuning runs—87.44% fewer than universal 125 × 10 retuning and 39.09% fewer than the conservative prospective schedule—versus 3768 for reliability-blind allocation. The guided configuration had the lower fresh-validation mean C_{max} on 16/24 targets, compared with 5/24 for blind allocation and three exact ties. Under the interpretive ±0.05% band, guided allocation was practically better on nine targets, blind allocation on 1, and 14 were practically tied. The mean target-level guided advantage was small (0.0503%) but positive, while the distribution of tuning effort differed substantially: guided search shifted runs away from GREEN and AMBER targets and toward RED targets. This supports the staged risk-guided procedure as an economical operational implementation within the tested panel, although the comparative evidence remains descriptive rather than confirmatory.

The RED subgroup illustrates how this comparison should be interpreted operationally. Guided allocation spends more of the fixed total budget in RED regions (1960 runs across four RED targets versus 628 under the reliability-blind schedule), yet the overall guided and blind totals remain essentially identical. On fresh validation, guided allocation is practically better on 2/4 RED targets and practically tied on 2/4 under the interpretive ±0.05% band, with no RED target practically favoring the blind allocation. This is descriptive evidence that concentrating effort in historically higher-risk regions can be efficient; it is not a claim that every individual RED retuning is economically warranted.

Stochastic selection uncertainty still limits what more tuning can guarantee. This is methodologically important because additional search produces a candidate selected under noisy evidence, not a deterministic improvement over the starting configuration. The relevant benefit to test is therefore lower out-of-sample loss or better allocation efficiency across targets, not target-wise dominance on every independent seed stream. The matched-budget experiment reinforces the need for independent validation: both methods choose among candidates using noisy confirmation means, and several target-level fresh-validation differences are very small. A future variance-aware configurator could use racing or sequential elimination to allocate replications to unresolved candidates, but the stopping rule would need to be fixed in advance and the final-validation stream kept separate.

The confirmatory statistics from the original prospective analysis should be read in the same spirit. They validate the risk map and conservative prospective implementation but do not establish policy superiority over direct center reuse; their non-significance also does not establish equivalence. The budget-matched evaluation addresses a different question and supports replacing the conservative replication schedule with the lower-cost staged implementation for operational use. The strongest demonstrated contribution is therefore the combination of prospectively validated risk ordering, a staged risk-guided policy with operational tuning-budget caps of ≤50, ≤150, and ≤500 runs for GREEN, AMBER, and

RED, respectively, matched-budget allocation evidence, and bounded quality loss within the tested panel—not a claim of universal dominance.

5.4. Generalizability, Reproducibility, and Directions for Extension

The generalizability of the numerical rule is deliberately bounded. The evidence concerns the fixed GA architecture and 5 × 5 × 5 parameter grid, makespan as the objective, the current generation-budget scheme, Taillard-based historical anchors, a Taillard-like discrete Uniform{1, ..., 99} processing-time regime for the prospective matrices, and interpolation only within n = 20–100 and m = 5–20. Different operators, elitism, initialization, local search, metaheuristics, objectives, or application loss functions should be treated as new calibration domains rather than as settings to which the present centers and thresholds are transferred unchanged.

Within-size instance variation is another important boundary. The prospective panel contains one independently generated matrix for each of 24 off-grid n × m size combinations. Repeated GA seeds measure algorithmic stochasticity on those matrices; they do not estimate the distribution of outcomes among multiple matrices of the same size. The 8/8 GREEN result is therefore evidence for this panel, not a precise future GREEN success probability. A stronger next-stage design should prespecify several independent matrices per selected size and use the matrix, not the stochastic GA run, as the unit of cross-instance inference.

The size-only rule also cannot detect structural distribution shift within the same n × m. The observed association between structural descriptors and response-surface sharpness suggests a promising extension, but the descriptors were not validated as action variables. A structure-aware policy should predefine the descriptors and prediction model on development data, freeze the mapping from predicted risk to action, and demonstrate incremental out-of-sample value over the size-only rule on an untouched multi-instance panel before any descriptor is allowed to change GREEN/AMBER/RED assignment.

The current scale envelope should likewise be extended by recalibration rather than extrapolation. Historical smoothness checks already show non-negligible interpolation error inside the present domain, with a mean absolute residual of 7.99 reliability percentage points and a maximum of 16.81 pp. For massive-scale PFSP, new anchors and multiple instances per anchor should be selected, the feasible parameter and generation-budget ranges should be recalibrated, and sparse, racing, or surrogate-guided configuration may replace exhaustive 125-configuration evaluation before a new transfer-risk map and new action thresholds are prospectively validated.

Prospective matrix provenance was verified by regenerating the original 24 targets using the prespecified PCG64DXSM seeds together with the recovered 16-bit integer draw detail. Independent checks reproduced 24/24 archived NEH values. The regenerated matrices, recovery audit, and verifier are provided in the Supplementary S1 and S2 repositories. This establishes seed-to-byte reproducibility of the prospective panel without changing the prespecified study inputs or outcomes.

The historical 100 × 20 reliability record was likewise verified through a comparison-level reconstruction. The canonical reconstruction reproduces the prespecified 595/840 reliability value, all 840 holdout comparisons are available for audit, and the exact ten-fold leave-one-instance-out range is 66.67–79.17%. A 597/840 result obtained under an alternate normalization is retained only as a sensitivity artifact and is not used as the canonical source for the reported anchor.

These limitations define the scope of the evidence without negating the prospective test itself. The transferable element of the study is the workflow—separate parameter recommendation from transfer-risk evidence, allocate tuning effort with a prespecified rule, lock the selected configuration, and evaluate it on independent data. Extending that

workflow to new algorithms, distributions, objectives, or scales requires new calibration and new prospective validation rather than direct numerical reuse of the present rule.

6. Conclusions

This study reframes GA parameter reuse in the PFSP as a decision about configuration effort. Historical experiments provide a transferable parameter center and a separate empirical reliability index. The final recommended policy maps that risk signal to increasing search breadth and staged replication: GREEN screens the center and axial neighbors, AMBER screens the local Cartesian neighborhood, and RED screens the full grid, with only leading candidates receiving confirmation replications.

The prospective evidence supports the risk-ranking part of this framework. Direct-center clipped loss increases from GREEN to AMBER to RED, and the continuous reliability score is negatively associated with direct-center loss. Under the primary 0.25-pp margin, both the policy and direct center reuse are within tolerance on 21/24 targets, so the prespecified tests do not establish statistical superiority. On the 16 AMBER+RED targets where additional tuning is actually invoked, however, mean clipped loss is 0.0786 pp for the policy versus 0.1483 pp for the center, and the stricter 0.10-pp margin yields 13/16 versus 7/16 within-tolerance outcomes. These intervention-only analyses are exploratory robustness evidence.

The original conservative prospective implementation required 6185 tuning runs, whereas the recommended staged operational policy required 3767 runs on the same 24-target panel. Relative to universal 30,000-run 125 × 10 retuning, this is an 87.44% reduction in configuration-run count and a 39.09% reduction relative to the conservative prospective schedule. These percentages are run-count comparisons rather than wall-clock savings because population size, generation depth, and problem dimensions differ across configurations and targets.

The experiments also identify clear limits. Full tuning under the original conservative prospective implementation rescues the 70 × 20 target but produces a marginal reverse ordering at 90 × 20, illustrating stochastic selection uncertainty. Five of eight GREEN response surfaces are completely tolerant at the primary margin, so flatness explains part of the original zero-tuning success, although two sharp GREEN cases remain consistent with informative center targeting. The staged budget-matched validation shows that the lower-cost operational policy can preserve broad full-grid coverage for RED while reducing replication burden: with essentially identical total tuning cost, guided allocation produces a lower fresh-validation mean on 16/24 targets and is practically better on 9/24 under the interpretive ±0.05% band, compared with 1/24 for blind allocation and 14/24 practical ties.

The demonstrated result is specific to the fixed GA architecture, C_{max} objective, Taillard-like processing-time regime, one prospective matrix per off-grid size, and the 20–100 job by 5–20 machine interpolation envelope. Different algorithms, processing-time distributions, objectives, or massive-scale instances require recalibration and new prospective validation. Structure-aware action assignment and adaptive replication are promising extensions only after their incremental value is established on untouched multi-instance data.

Reproducibility is strengthened by the archived analyses, prespecified actions, summary outcomes, reference implementation, reproducibility checks, canonical 100 × 20 595/840 comparison-level reconstruction, and exact recovery of the original 24 prospective matrices. The matrix recovery reproduces all 24 archived values. The recommended staged policy and its budget-matched validation are supplied with the explicit selection protocol, selected configurations, 960 fresh validation observations, and cross-package verification utilities.

The resulting scientific claim is deliberately bounded: historical transfer evidence can be used to rank reuse risk and to allocate GA tuning effort selectively within the tested PFSP domain. The original prospective experiment validates the risk-ranking and increasing-search-breadth logic, while the budget-matched experiment supports the lower-cost staged implementation, summarized by operational tuning-budget caps of $\leq 50/\leq 150/\leq 500$ runs, as the recommended policy on this panel. The study does not establish universal dominance across targets, algorithms, instance distributions, or tuning strategies.

Supplementary Materials: The supporting computational and data materials for this study are publicly available in two dedicated GitHub repositories. Supplementary S1: Code and Reproducibility is available at https://github.com/AhmetCihanDuzce/SupplementaryS1_Reliability-Guided-Allocation-of-Genetic-Algorithm-Tuning-Effort-in-PFSS (Access date: 25 October 2026) and contains the computational specifications, reference GA implementation, deterministic seed rules, exact recovery and verification utilities for the 24 prospective matrices, executable parameter-grid and parameter-selection workflows, the original prospective validation protocol, and the staged operational policy with its budget-matched validation protocol. Supplementary S2: Data, Results, and Quality Assurance is available at https://github.com/AhmetCihanDuzce/SupplementaryS2_Reliability-Guided-Allocation-of-Genetic-Algorithm-Tuning-Effort-in-PFSS (Access date: 25 October 2026) and contains the historical calibration and transfer-validation records underlying the nine-anchor reliability map, the complete canonical 100 × 20 595/840 reconstruction, the complete target-level results and quality-assurance material for the original 24-target prospective validation, and the 24-target staged-policy-versus-blind comparison with 960 fresh common-random-number validation observations and audit files.

Author Contributions: Conceptualization, A.F. and N.F.; methodology, A.F. and A.C.; formal analysis, G.A.K.; investigation, P.Ş.Ç.; data curation, A.C.; writing—original draft preparation, N.F. and A.F.; writing—review and editing, A.F., A.C., G.A.K. and P.Ş.Ç.; visualization, A.C.; supervision, N.F.; resources, N.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research and the APC were funded by the Kocaeli University Scientific Research Projects Coordination Unit under Project Code FBA-2026-5184 (Project ID: 5184).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data, code, and computational materials supporting this study are publicly available in two dedicated GitHub repositories. Supplementary S1: Code and Reproducibility is available at https://github.com/AhmetCihanDuzce/SupplementaryS1_Reliability-Guided-Allocation-of-Genetic-Algorithm-Tuning-Effort-in-PFSS (Access date: 25 October 2026) and includes the experimental specifications, portable reference GA implementation, deterministic GA and matrix seed rules, retained input instances, parameter-grid and parameter-selection workflows, the original prospective protocol, exact prospective-matrix recovery utilities, the staged operational policy and budget-matched validation protocol, and verification scripts. Supplementary S2: Data, Results, and Quality Assurance is available at https://github.com/AhmetCihanDuzce/SupplementaryS2_Reliability-Guided-Allocation-of-Genetic-Algorithm-Tuning-Effort-in-PFSS (Access date: 25 October 2026) and includes the retained historical transfer records, canonical 100 × 20 reconstruction, target-level prospective results, robustness and sensitivity analyses, staged-policy selected configurations, 960 fresh validation observations, and quality-assurance material.

Acknowledgments: The authors thank the Kocaeli University Scientific Research Projects Coordination Unit for financial support.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Garey, M.R.; Johnson, D.S.; Sethi, R. The complexity of flowshop and jobshop scheduling. *Math. Oper. Res.* **1976**, *1*, 117–129. [CrossRef]
2. Taillard, E. Benchmarks for basic scheduling problems. *Eur. J. Oper. Res.* **1993**, *64*, 278–285. [CrossRef]
3. Ruiz, R.; Maroto, C. A comprehensive review and evaluation of permutation flowshop heuristics. *Eur. J. Oper. Res.* **2005**, *165*, 479–494. [CrossRef]
4. Fernandez-Viagas, V.; Sanchez-Mediano, L.; Angulo-Cortes, A.; Gomez-Medina, D.; Molina-Pariente, J.M. The permutation flow shop scheduling problem with human resources: MILP models, decoding procedures, NEH-based heuristics, and an iterated greedy algorithm. *Mathematics* **2022**, *10*, 3446. [CrossRef]
5. Amirghasemi, M.; Voß, S.; Garn, W.; Arjomandi, A.; Ogie, R. A new one-machine decomposition technique for solving the permutation flow-shop scheduling problem. *Algorithms* **2026**, *19*, 574. [CrossRef]
6. Eiben, A.E.; Hinterding, R.; Michalewicz, Z. Parameter control in evolutionary algorithms. *IEEE Trans. Evol. Comput.* **1999**, *3*, 124–141. [CrossRef]
7. Eiben, A.E.; Smit, S.K. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm Evol. Comput.* **2011**, *1*, 19–31. [CrossRef]
8. Reeves, C.R. A genetic algorithm for flowshop sequencing. *Comput. Oper. Res.* **1995**, *22*, 5–13. [CrossRef]
9. Ruiz, R.; Maroto, C.; Alcaraz, J. Two new robust genetic algorithms for the flowshop scheduling problem. *Omega* **2006**, *34*, 461–476. [CrossRef]
10. Hutter, F.; Hoos, H.H.; Leyton-Brown, K.; Stützle, T. ParamILS: An automatic algorithm configuration framework. *J. Artif. Intell. Res.* **2009**, *36*, 267–306. [CrossRef]
11. López-Ibáñez, M.; Dubois-Lacoste, J.; Cáceres, L.P.; Birattari, M.; Stützle, T. The irace package: Iterated racing for automatic algorithm configuration. *Oper. Res. Perspect.* **2016**, *3*, 43–58. [CrossRef]
12. Birattari, M.; Stützle, T.; Paquete, L.; Varrentrapp, K. A racing algorithm for configuring metaheuristics. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002), New York, NY, USA, 9–13 July 2002*; Morgan Kaufmann: San Francisco, CA, USA, 2002; pp. 11–18.
13. Hutter, F.; Hoos, H.H.; Leyton-Brown, K. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization*; Coello Coello, C.A., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6683, pp. 507–523. [CrossRef]
14. Zhou, T.; Zhang, W.; Niu, B.; He, P.; Yue, G. Parameter control framework for multiobjective evolutionary computation based on deep reinforcement learning. *Int. J. Intell. Syst.* **2024**, *2024*, 6740701. [CrossRef]
15. Qi, X.; Zhang, Z.; Zhang, J.; Song, Y. A hybrid Kepler optimization algorithm based on Q-learning for solving permutation flow shop scheduling problems. *IEEE Access* **2026**, *14*, 78242–78259. [CrossRef]
16. Ma, Z.; Guo, H.; Gong, Y.-J.; Zhang, J.; Tan, K.C. Toward automated algorithm design: A survey and practical guide to Meta-Black-Box-Optimization. *IEEE Trans. Evol. Comput.* **2026**, *30*, 667–687. [CrossRef]
17. Xue, X.; Yang, C.; Feng, L.; Zhang, K.; Song, L.; Tan, K.C. Solution transfer in evolutionary optimization: An empirical study on sequential transfer. *IEEE Trans. Evol. Comput.* **2024**, *28*, 1776–1793. [CrossRef]
18. Zou, P.; Rajora, M.; Liang, S.Y. Multimodal optimization of permutation flow-shop scheduling problems using a clustering-genetic-algorithm-based approach. *Appl. Sci.* **2021**, *11*, 3388. [CrossRef]
19. Lu, Y.; Yuan, Y.; Yasenjiang, J.; Sitahong, A.; Chao, Y.; Wang, Y. An optimized method for solving the green permutation flow shop scheduling problem using a combination of deep reinforcement learning and improved genetic algorithm. *Mathematics* **2025**, *13*, 545. [CrossRef]
20. Dong, Z.; Ren, T.; Weng, J.; Qi, F.; Wang, X. Minimizing the late work of the flow shop scheduling problem with a deep reinforcement learning based approach. *Appl. Sci.* **2022**, *12*, 2366. [CrossRef]
21. Fığlalı, N.; Özkale, C.; Engin, O.; Fığlalı, A. Investigation of Ant System parameter interactions by using design of experiments for job-shop scheduling problems. *Comput. Ind. Eng.* **2009**, *56*, 538–559. [CrossRef]
22. Sánchez Márquez, E.M.; Pérez-Rodríguez, R.; Ornelas-Rodríguez, M.; Puga-Soberanes, H.J. An evolutionary strategy based on the Generalized Mallows model applied to the mixed no-idle permutation flow shop scheduling problem. *Math. Comput. Appl.* **2025**, *30*, 39. [CrossRef]
23. Rice, J.R. The algorithm selection problem. *Adv. Comput.* **1976**, *15*, 65–118. [CrossRef]
24. Kadioglu, S.; Malitsky, Y.; Sellmann, M.; Tierney, K. ISAC—Instance-specific algorithm configuration. In *ECAI 2010: 19th European Conference on Artificial Intelligence*; Coelho, H., Studer, R., Wooldridge, M., Eds.; Frontiers in Artificial Intelligence and Applications; IOS Press: Amsterdam, The Netherlands, 2010; Volume 215, pp. 751–756. [CrossRef]
25. Smith-Miles, K.A. Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Comput. Surv.* **2009**, *41*, 1–25. [CrossRef]

26. Kerschke, P.; Hoos, H.H.; Neumann, F.; Trautmann, H. Automated algorithm selection: Survey and perspectives. *Evol. Comput.* **2019**, *27*, 3–45. [CrossRef] [PubMed]
27. Nawaz, M.; Enscore, E.E., Jr.; Ham, I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega* **1983**, *11*, 91–95. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.