Informative Decision Situation Selection for Phenotypic Characterization-Based Surrogate-Assisted Genetic Programming in Dynamic Container Terminal Truck Scheduling

Leshan Tan[1][0009−0002−6060−3834], Chenwei Jin[1][0009−0003−5694−9709], Xinan Chen[1][0000−0001−9620−3264], Rong Qu[2][0000−0001−8318−7509], and Ruibin Bai[1⋆][0000−0003−1722−568X]

[1] University of Nottingham, Ningbo, China
{leshan.tan, chenwei.jin, xinan.chen, ruibin.bai}@nottingham.edu.cn
[2] University of Nottingham, Nottingham, United Kingdom
rong.qu@nottingham.ac.uk

Abstract. Surrogate-assisted genetic programming (SAGP) addresses the computational challenge of evolving heuristics for dynamic scheduling by replacing expensive fitness evaluations with learned approximations. However, standard surrogate models require fixed-size inputs, whereas GP individuals are variable-sized symbolic programs, making characterization a core challenge. Phenotypic characterization (PC) observes each individual's behavior over a selected set of decision situations (DSs), where each DS captures a scheduling decision moment and its candidate actions. The selected DSs therefore determine what behavioral evidence is available to the surrogate model. Existing studies typically use fixed DS sets that are randomly sampled or left unspecified, leaving the construction of informative DS sets under a fixed observation budget as an open question. In this paper, we study informative DS selection for PC-based SAGP in dynamic container terminal truck scheduling (DCTTS). We develop a multi-stage framework that first collects a broad raw DS pool using committee policies and constructs a representative candidate pool, then applies Alignment-Guided Quality-Coverage Greedy (AQCG) to select complementary DSs within this observation budget. AQCG measures DS informativeness by quantifying how well each DS's induced behavioral distinctions align with true performance differences, then greedily maximizes set-level coverage of this alignment information. Experimental results on DCTTS show that AQCG achieves the best overall scheduling performance among the compared methods and improves surrogate guidance over random sampling and quality-only selection.

Keywords: Genetic Programming · Surrogate Model · Phenotypic Characterization · Decision Situation · Dynamic Container Terminal Truck Scheduling

⋆ Corresponding author.

1 Introduction

Genetic programming (GP) [14] has been successfully adopted to evolve heuristics for dynamic scheduling problems [1,3,10]. However, when fitness evaluation relies on stochastic simulation, the computational cost becomes prohibitive, limiting GP's applicability to real-world scenarios. This occurs as each candidate rule requires evaluation across multiple independent simulation replications [7,8,19], and a GP run evaluates thousands of individuals over many generations making the cumulative simulation workload a major bottleneck. To address this, surrogate-assisted genetic programming (SAGP) uses predictive models to approximate fitness values, substantially reducing computational cost [9,13,17,25].

Despite its success, SAGP effectiveness critically depends on how GP individuals are characterized for surrogate modeling. Standard surrogate models are designed to operate on fixed-length numerical feature vectors [9,13], whereas GP individuals are symbolic programs with variable structures (e.g. trees [15]). This fundamental structural mismatch necessitates an intermediate characterization step that transforms each individual into a fixed-dimensional representation before surrogate modeling can be applied. Consequently, both the predictive model and the characterization informativeness determine surrogate quality.

One widely adopted characterization approach is phenotypic characterization (PC), which builds this representation by observing how an individual behaves across a collection of decision situations (DSs) rather than relying on syntactic program features [9]. In dynamic scheduling contexts, each DS represents a specific online decision moment with the candidate actions available for selection. The PC is then constructed by recording how the dispatching rule evaluates, ranks, or selects among these candidates across multiple DSs. The quality of the resulting PC therefore depends critically on which DSs are included in the observation set, as different DS selections capture different aspects of behavioral variation and consequently determine whether the PC can effectively distinguish individuals with different behaviors and performance.

Despite the importance of DS selection, most studies related to PC do not treat it as a primary research focus. The typical approach is to sample DSs randomly from a pre-collected pool, or to leave the construction procedure unspecified. Instead, research has concentrated on surrogate model design, surrogate management strategies, or the characterization methods used to represent behavior on a given DS set [9,17,25]. This gap is important to address, as DS selection propagates through the entire SAGP pipeline. A DS set that is redundant, narrow, or uninformative degrades characterization quality, which in turn reduces surrogate accuracy and distorts selection pressure during search, ultimately harming algorithm performance. Given that observation budgets are limited in practice, improving DS selection offers a direct way to enhance SAGP without modifying the surrogate model or characterization method.

However, constructing an informative DS set under a fixed observation budget presents technical challenges. The space of candidate DSs grows rapidly with problem complexity: each decision moment during simulation constitutes a potential observation point, leading to thousands of candidate DSs even for moderately sized problems. More critically, DSs exhibit complex dependencies: two DSs may appear distinct in their candidate-feature configurations yet elicit nearly identical behavioral responses from most individuals, making them functionally redundant. Detecting this redundancy requires evaluating individual behavior on each DS, not merely comparing candidate features. Consequently, effective DS selection involves balancing two competing objectives: the selected set should cover diverse behavioral patterns to maximize characterization informativeness, while remaining compact to satisfy the fixed observation budget.

This paper investigates informative DS selection under a fixed observation budget for PC-based SAGP in DCTTS. The main contributions are as follows:

- We formulate informative DS selection under a fixed observation budget as a primary problem in PC-based SAGP, shifting the DS set from a fixed preprocessing choice, often randomly sampled or unspecified, to an explicit optimization target.
- We develop a multi-stage DS selection framework that collects a broad raw DS pool using committee policies, constructs a representative candidate pool through state-based and behavior-based deduplication, and selects the final DS set with AQCG. AQCG scores DSs by aligning induced behavioral distinctions with true performance gaps, then greedily maximizes coverage of these pairwise alignment contributions.
- We conduct comprehensive experiments on DCTTS, showing that AQCG achieves the best overall performance among the compared methods and improves surrogate guidance over random sampling and quality-only selection baselines, demonstrating the effectiveness of coverage-aware DS selection.

2 Background and Related Work

2.1 Surrogate-Assisted Genetic Programming

Genetic Programming (GP) [14] has emerged as a prominent hyper-heuristic approach [2,18,24] for evolving dispatching rules in dynamic scheduling environments [1,5,6,23]. These settings belong to a broader family of online vehicle-assignment and pickup-and-delivery problems that require adaptive dispatch policies under uncertainty [12,20]. In GP-based scheduling applications, GP individuals are represented as variable-length symbolic priority functions that map attributes of available actions in a decision situation to priority scores. This symbolic encoding offers key advantages: representational flexibility that allows discovering complex decision logic, and interpretability that enables practitioners to inspect and validate resulting rules. In uncertain dynamic environments, evaluating such rules through repeated simulations makes fitness evaluation the dominant computational bottleneck [6,8,9,21].

Surrogate-Assisted Genetic Programming (SAGP) mitigates this bottleneck by using computationally inexpensive predictive models to approximate fitness values [9,13,17,25]. However, standard surrogate models operate on fixed-length numerical feature vectors, whereas GP individuals are variable-sized symbolic programs. This structural mismatch necessitates a characterization mechanism that transforms GP individuals into fixed-dimensional representations for surrogate modeling, with phenotypic characterization being a widely adopted solution.

Phenotypic Characterization and Decision Situation Phenotypic Characterization (PC) constructs fixed-dimensional representations by observing how each GP individual, viewed as a dispatching rule, behaves across a collection of Decision Situations (DSs) [9]. In dynamic scheduling, each DS captures a decision moment as an ordered list of candidate actions, each described by terminal-feature values available to dispatching policies. The PC records how the target rule ranks or selects among these candidates. Table 1 gives an illustrative calculation. In each DS, the target GP rule selects the candidate task with the lowest score, and the PC element records the selected task's rank under the manual reference policy. The resulting PC vector is therefore $(1, 3, 2)$.

Table 1. Illustrative example of PC construction.

DS	Candidate	Ref. rank(score)	GP rank(score)	PC element
d_1	t_1	**1(0.30)**	**1(0.20)**	**1**
	t_2	2(0.55)	3(0.90)	
	t_3	3(0.80)	2(0.60)	
d_2	t_4	**3(0.90)**	**1(0.10)**	**3**
	t_5	1(0.30)	2(0.40)	
	t_6	2(0.55)	3(0.70)	
d_3	t_7	1(0.20)	2(0.60)	**2**
	t_8	**2(0.45)**	**1(0.30)**	
	t_9	3(0.80)	3(0.90)	

To maintain computational efficiency, the DS set is kept much smaller than the number of DSs encountered during full simulation, creating a fixed observation budget, i.e., a limit on the number of DSs used to observe rule behavior. Most prior work has focused on surrogate model design, surrogate management strategies, or characterization methods [9,17,21,25], often leaving DS selection fixed or implicit, e.g., by random sampling. This motivates optimizing which DSs are selected under a fixed observation budget, a complementary perspective readily incorporated into existing PC-based SAGP approaches.

Preselection Strategy Preselection is a SAGP framework that has proven effective in scheduling domains [9,21,22]. As illustrated in Figure 1, preselection begins by fully evaluating an initial population and using these evaluated individuals to build the surrogate. In each subsequent generation, parent selection

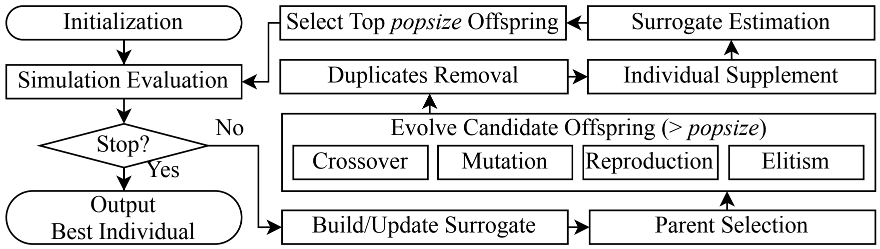

Fig. 1. Preselection-based SAGP framework.

and genetic operators generate an oversized offspring pool. Duplicate removal is then applied to reduce repeated candidates, and individual supplement is used when additional candidates are needed. The surrogate estimates the fitness of the resulting candidate offspring based on their PC-based feature vectors, and only the top-ranked subset undergoes full simulation evaluation. After exact evaluation, the surrogate is updated using all fully evaluated individuals. This strategy enables the search to examine more genetic variations while reducing expensive simulation evaluations. However, poor surrogate predictions can misguide preselection and degrade algorithm performance, making the quality of PC-based characterization critical to preselection effectiveness.

One commonly used surrogate for PC-based SAGP is 1-nearest neighbor (1NN). Let $\mathcal{A}$ denote the archive of fully evaluated individuals. For an unevaluated individual x, the selected DS set is used to compute its PC vector. The nearest stored sample is identified in the PC space by

$$z^\star = \arg\min_{z \in \mathcal{A}} \mathrm{Dist}_{\mathrm{PC}}(x, z),$$

where $\mathrm{Dist}_{\mathrm{PC}}(x, z)$ measures the distance between their PC vectors, e.g., using Euclidean distance. Under the nearest-neighbor assumption, individuals with similar PC vectors are expected to have similar fitness values, so the predicted fitness is assigned as $\hat{F}(x) = F(z^\star)$. As more individuals undergo exact simulation evaluation, they are added to $\mathcal{A}$ and become available for later predictions.

2.2 Dynamic Container Terminal Truck Scheduling

A typical container terminal consists of berth-side quay cranes that handle ship loading and unloading operations, yard-side storage blocks managed by yard cranes, and entry-exit gates connecting to external logistics networks. Internal trucks shuttle containers among these facilities under operational constraints. Since trucks coordinate the interactions among the terminal's principal resources, effective dispatching decisions are central to overall terminal performance. Dynamic Container Terminal Truck Scheduling (DCTTS) addresses the real-time assignment of trucks to transport tasks that arrive dynamically during terminal operations [3,4,11], with the objective of maximizing throughput (measured in TEUs/h) under stochastic service times and uncertain travel conditions.

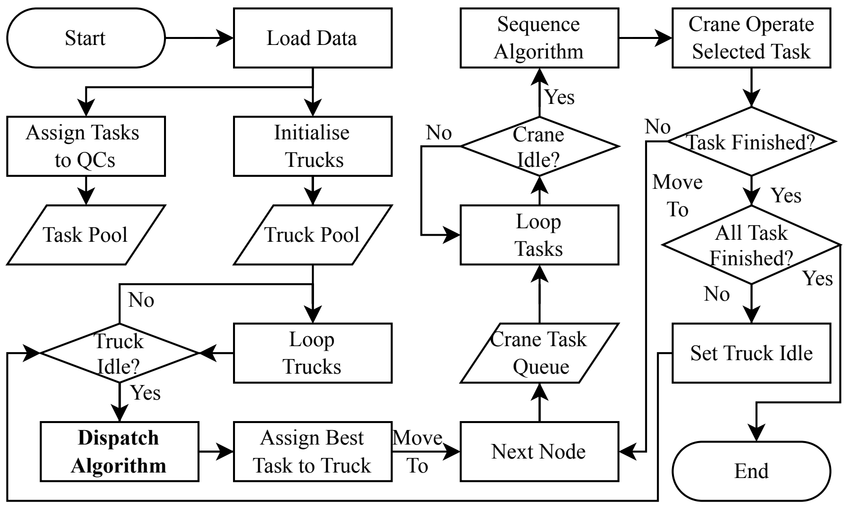

Fig. 2. Event-driven dispatch workflow in DCTTS (adapted from [17]).

DCTTS operates through an event-driven model, as illustrated in Figure 2. Whenever a truck becomes idle, the system generates a decision situation (DS) and invokes the dispatch algorithm to select from feasible candidate tasks in the task pool. The dispatch algorithm applies a policy, either a manual heuristic or a GP rule, to compute a priority score for each candidate based on terminal features, then selects the highest-priority task. In this paper, lower scores indicate higher priority. As trucks complete tasks and new transport requests arrive, the number of candidates varies across DSs. For GP rules, the priority-scoring behavior observed over multiple DSs reflects their operational characteristics and serves as the foundation for phenotypic characterization in SAGP.

3 Methodology

3.1 Multi-Stage Framework Overview

Motivated by the need to select compact yet informative DS sets, our approach decomposes DS selection into three stages with explicit inputs and outputs. The first stage emphasizes breadth: committee policies are executed on the training instances to collect a broad raw DS pool. The second stage controls redundancy: state-based and behavior-based deduplication remove repeated or functionally equivalent observations and transform the raw pool into a representative candidate pool. The third stage enforces the observation budget: AQCG selects the final compact DS set from the representative pool. Figure 3 illustrates this pipeline, where different shapes and colors represent DS diversity in terms of observable decision states and behavioral characteristics.

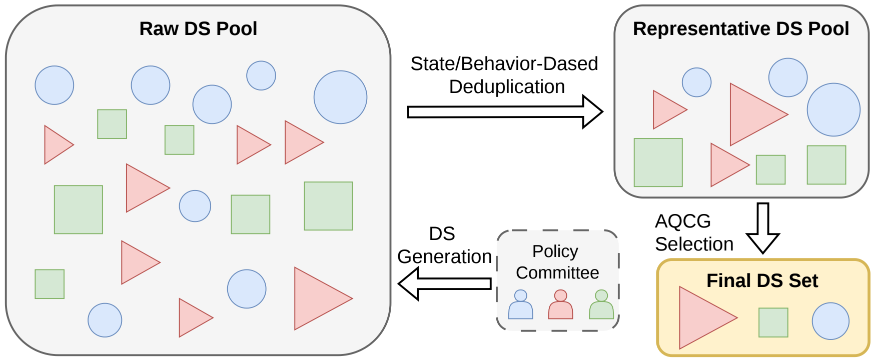

Fig. 3. Multi-stage DS selection framework from raw DS generation to representative-pool construction and final AQCG selection.

Let $\mathcal{D}_{\text{raw}}$ denote the raw DS pool and $\mathcal{D}_{\text{rep}}$ the representative candidate pool after deduplication. Under observation budget B, $\mathcal{D}_{\text{final}}$ denotes the DS set used for surrogate-assisted search, where $|\mathcal{D}_{\text{final}}| = B$. The goal is to construct an informative $\mathcal{D}_{\text{final}}$ for downstream surrogate modeling and search.

3.2 Committee-Driven DS Generation

The raw DS pool $\mathcal{D}_{\text{raw}}$ is generated offline by executing the committee policies specified in Section 4.4 over the training instances. The committee is designed to exhibit diverse behavioral patterns by selecting policies that emphasize different operational priorities. This behavioral diversity is critical, as it exposes DSs where different policies make contrasting decisions, thereby providing a rich foundation for subsequent selection of informative DSs.

3.3 Representative Candidate Pool Construction

The raw DS pool $\mathcal{D}_{\text{raw}}$ generated by the committee may contain substantial redundancy. Multiple policies may visit the same observable decision state at different times, or different trajectories may produce functionally equivalent DSs despite minor differences. To construct the representative candidate pool $\mathcal{D}_{\text{rep}}$, we apply state-based and behavior-based deduplication. DSs with fewer than two candidates are discarded because they provide no ranking information.

State-Based Deduplication The first step removes identical observable decision states. Two DSs are exact duplicates if they have the same candidate sequence and matching feature vector for each candidate. Candidate order is retained because it affects tie-breaking between equally scored tasks. For each duplicate group, we retain one DS and record its provenance, including occurrence counts, source policies, and originating instances.

Behavior-Based Deduplication The second step removes behavioral redundancy: distinct observable states that induce identical responses. For each DS, we replay all committee policies and record their candidate-ranking matrix, with one row per policy, using the same priority and tie-breaking convention. DSs with the same candidate count and ranking matrix are grouped. From each group, we retain the most frequently observed DS, using source-policy coverage and original appearance order only as tie-breakers. The resulting representatives are behaviorally distinct and supported by diverse trajectories.

3.4 Alignment-Guided Quality-Coverage Greedy (AQCG)

Given the representative candidate pool $\mathcal{D}_{\text{rep}}$, constructing the final set $\mathcal{D}_{\text{final}}$ under observation budget B (the maximum number of DSs to select) requires balancing individual DS informativeness with set-level complementarity. A DS with high individual informativeness may still be redundant with other selected DSs, repeatedly emphasizing the same policy-pair contrasts while leaving other contrasts undercovered. Such redundancy limits the characterization's ability to capture the full spectrum of behavioral variation, which degrades surrogate prediction accuracy and ultimately undermines search performance.

We introduce **Alignment-Guided Quality-Coverage Greedy (AQCG)**, a greedy set construction method for DS selection. Distance-based surrogate models assume that behavioral similarity should correlate with performance similarity [9,13]. Building on this assumption, we measure DS informativeness through *alignment*: the degree to which the behavioral distinctions induced by a DS on the committee align with the true performance differences among committee policies. A DS with high alignment provides more informative characterization features for surrogate learning. AQCG decomposes each DS into its pairwise alignment contributions across all policy pairs, then greedily constructs the final set by iteratively selecting DSs that maximally augment the current set's coverage of these alignment contributions.

Alignment Metric Formulation Let $\mathcal{P} = \{\pi_1, \ldots, \pi_K\}$ denote the committee of dispatching policies used to generate the raw DS pool. For each policy π_i, let $\overline{\text{Obj}}_i$ denote its mean throughput over the training set. For each policy pair (i, j) with $1 \leq i < j \leq K$, define the performance gap as:

$$G_{ij} = |\overline{\text{Obj}}_i - \overline{\text{Obj}}_j|.$$

For each DS $d \in \mathcal{D}_{\text{rep}}$ and policy pair (i, j), let $\Delta_{ij}(d)$ denote the behavioral disagreement induced by d between policies π_i and π_j. We compute the Spearman rank correlation [16] $\rho_{ij}(d)$ between the two policies' rankings of candidates in d, and transform it to a disagreement measure via

$$\Delta_{ij}(d) = \frac{1 - \rho_{ij}(d)}{2} \in [0, 1].$$

Thus, identical rankings give $\Delta_{ij}(d) = 0$, while more different rankings produce larger disagreement values.

To place performance gaps on the same $[0, 1]$ scale as behavioral disagreement, we apply min-max normalization:

$$\tilde{G}_{ij} = \frac{G_{ij} - \min_{i<j} G_{ij}}{\max_{i<j} G_{ij} - \min_{i<j} G_{ij}}.$$

The pairwise alignment contribution is then defined as:

$$a_{ij}(d) = \max\left(0, 1 - \left|\Delta_{ij}(d) - \tilde{G}_{ij}\right|\right).$$

This formulation compares the local behavioral separation induced by d, represented by $\Delta_{ij}(d)$, with the global performance difference between the two policies, represented by $\tilde{G}_{ij}$. When $\Delta_{ij}(d) \approx \tilde{G}_{ij}$, the DS distinguishes the policy pair to a degree consistent with their true performance gap, yielding $a_{ij}(d) \approx 1$. Conversely, when the two quantities are far apart, $a_{ij}(d)$ approaches 0. Thus, $a_{ij}(d)$ quantifies how well d provides alignment evidence for pair (i, j).

The average pairwise alignment score for DS d is then:

$$q_{\text{align}}(d) = \frac{1}{K(K-1)/2} \sum_{i<j} a_{ij}(d).$$

The score $q_{\text{align}}(d)$ quantifies overall alignment quality by averaging pairwise contributions. AQCG uses this score to choose the initial DS and to prefer higher-quality candidates when marginal gains are tied, while optimizing for set-level coverage rather than individual DS scores.

Set-Level Coverage Objective AQCG reframes DS selection as a set-level coverage problem. For a given set $S \subseteq \mathcal{D}_{\text{rep}}$, define the current best-covered alignment contribution for each policy pair (i, j) as:

$$c_{ij}(S) = \max_{d' \in S} a_{ij}(d').$$

Here, d' denotes one of the DSs already selected in S. This quantity represents the strongest alignment evidence that the current set S has achieved for pair (i, j). If S already contains a DS with strong evidence for this pair, then $c_{ij}(S)$ is high. Otherwise, (i, j) remains poorly covered.

For any candidate DS $d \notin S$, the marginal gain of adding d to S is:

$$\Delta J(d \mid S) = \sum_{i<j} \max\left(0, a_{ij}(d) - c_{ij}(S)\right).$$

This formulation explicitly rewards complementary alignment information. If d provides stronger alignment evidence than the current set for some pair (i, j), it contributes positive marginal gain. If d merely repeats information already covered by S, it contributes zero marginal gain. By maximizing marginal gain at each step, AQCG prioritizes DSs that augment the set's coverage rather than redundantly reinforcing already well-covered pairs.

Greedy Construction Algorithm AQCG constructs the final DS set $\mathcal{D}_{\text{final}}$ through a greedy forward selection procedure. The algorithm proceeds as follows:

Algorithm 1 AQCG: Alignment-Guided Quality-Coverage Greedy Selection

Require: Candidate pool $\mathcal{D}_{\text{rep}}$, budget B, committee $\mathcal{P}$
Ensure: Final DS set $\mathcal{D}_{\text{final}}$
$S \leftarrow \emptyset$
$d_{\text{init}} \leftarrow \arg\max_{d \in \mathcal{D}_{\text{rep}}} q_{\text{align}}(d)$
$S \leftarrow S \cup \{d_{\text{init}}\}$
Update $c_{ij}(S) \leftarrow a_{ij}(d_{\text{init}})$ for all (i, j)
for $k = 2$ to B **do**
  for each $d \in \mathcal{D}_{\text{rep}} \setminus S$ **do**
    Compute $\Delta J(d \mid S) = \sum_{i<j} \max(0, a_{ij}(d) - c_{ij}(S))$
  end for
  Select d^* with maximum $\Delta J(d \mid S)$, breaking ties by larger $q_{\text{align}}(d)$
  $S \leftarrow S \cup \{d^*\}$
  Update $c_{ij}(S) \leftarrow \max_{s \in S} a_{ij}(s)$ for all (i, j)
end for
return $\mathcal{D}_{\text{final}} = S$

Computational Complexity AQCG's greedy construction requires $O(B \cdot K^2 \cdot |\mathcal{D}_{\text{rep}}|)$ operations, where B is the budget, K is the committee size, and $|\mathcal{D}_{\text{rep}}|$ is the candidate pool size. Pairwise contributions $a_{ij}(d)$ are precomputed, so each iteration evaluates marginal gains for remaining candidates in $O(K^2 \cdot |\mathcal{D}_{\text{rep}}|)$ time, and tie-breaking by q_{align} does not change the bound. This one-time offline preprocessing is completed prior to training.

4 Experimental Design

4.1 Simulation Environment and Datasets

All experiments are conducted using the DCTTS simulator described in Section 2.2. We consider eight operating scenarios obtained from the Cartesian product of three factors: the workload size $n \in \{200, 400\}$ tasks, the loading-task ratio $r \in \{0.3, 0.7\}$, and the number of trucks per quay crane $\kappa \in \{5, 8\}$. We denote each scenario by $\langle n, r, \kappa \rangle$. Each scenario contains 10 training instances and 100 testing instances. The mixed training and testing sets combine instances from all eight scenarios, containing 80 training instances and 800 testing instances.

4.2 Fitness Evaluation

The fitness of a dispatching rule is measured by its average relative improvement over the reference policy on the training set, with a tree-size penalty [17]:

$$F(h) = \frac{1}{|\mathcal{I}_{\text{train}}|} \sum_{m \in \mathcal{I}_{\text{train}}} \frac{\text{Obj}(m, h) - \text{Obj}(m, \text{ref})}{\text{Obj}(m, \text{ref})} - \lambda \cdot S(h), \tag{1}$$

where $\mathcal{I}_{\text{train}}$ contains the 80 training instances, $\text{Obj}(m, h)$ is the throughput achieved by rule h on instance m, ref denotes the manual reference policy, $S(h)$ is the number of nodes in the GP tree, and λ controls the bloat penalty. Higher $F(h)$ values are better because the objective is throughput maximization.

4.3 Comparison Protocol

To isolate the effect of DS selection, all SGP-PS variants use the same preselection backbone, 1NN surrogate, PC construction procedure, and final DS budget B. They differ only in how the DS set is constructed. The backbone follows the preselection framework described in Section 2.1 with PC-based characterization [9]. We compare the following SGP-PS variants:

- **Random**: uniformly samples B DSs from the raw DS pool before representative preprocessing.
- **RandomRep**: uniformly samples B DSs from the representative candidate pool after state-based and behavior-based deduplication.
- **QAlign**: selects the top B DSs by the individual q_{align} score, testing quality-only selection without set-level coverage.
- **AQCG**: selects B DSs from the representative candidate pool by greedily maximizing set-level coverage of pairwise alignment contributions.

Standard GP, which performs simulation-based fitness evaluation without a surrogate, is included as a non-surrogate baseline for scheduling performance. Surrogate-quality metrics are reported only for SGP-PS variants.

To account for DS-set randomness, random baselines evaluate independently sampled DS sets, with 30 independent runs per set. Their results are averaged by seed to obtain 30 seed-level values for reporting. QAlign, AQCG, and GP are each evaluated with 30 independent runs.

4.4 Parameter Settings

The same settings as in [17] are used for all GP-based methods. The population size is 500, tournament size is 5, the number of elites is 10, the crossover, mutation, and reproduction rates are 0.8, 0.15, and 0.05, the maximum tree depth is 10, and the penalty factor is 10^{-7}. A 5 h training budget is used, approximately the time required by standard GP to complete 50 generations.

The terminal set includes 11 truck-dispatch features covering travel time, crane workload, task type, dispatch type, endpoint queue sizes, target-truck counts, remaining crane tasks, and average pickup and drop-off times. The function set comprises logical, conditional, comparison, and arithmetic primitives: `AND`, `OR`, `IF`, $\leq$, $\geq$, max, min, $+$, $-$, $\times$, and protected division.

For SAGP-specific parameters, the DS budget is set to $B = 40$ and the preselection brood size is 2. For DS generation, the committee contains the manual reference policy and four opposite heuristic pairs, as listed in Table 2.

Table 2. Committee policies used for DS generation.

Criterion	Policies	Priority rule
Manual	Manual	Expert-designed reference policy
Travel time	STT / LTT	Shortest / longest travel time
Crane workload	MTR / STR	Most / fewest remaining crane tasks
Queue pressure	LCQ / HCQ	Lowest / highest queue pressure
Processing time	SPT / LPT	Shortest / longest processing time

These opposite pairs expose contrasting dispatching preferences and diversify the raw DS pool. Each committee policy is used for both main-task and sub-task dispatching decisions, while crane dispatching follows first-come-first-served (FCFS).

5 Results and Discussion

For all reported metrics, larger values indicate better performance or surrogate quality. Statistical comparisons between AQCG and each non-AQCG method use the Wilcoxon rank-sum test at $\alpha = 0.05$. The symbols (+), (−), and (=) indicate that AQCG is significantly better than, significantly worse than, and not significantly different from the compared method, respectively.

5.1 Algorithm Performance

Table 3. Mean (std) test performance (×100).

		SGP-PS			
Scenario	GP	Random	RandomRep	QAlign	AQCG
Overall	6.30(0.31)(+)	6.50(0.23)(+)	6.52(0.22)(+)	6.31(0.34)(+)	**6.70(0.23)**
$\langle 200, 0.3, 5\rangle$	7.35(0.60)(+)	7.83(0.61)(+)	7.97(0.42)(=)	7.41(0.83)(+)	**8.23(0.65)**
$\langle 200, 0.3, 8\rangle$	1.88(0.38)(+)	2.22(0.38)(=)	2.17(0.24)(=)	1.87(0.73)(+)	**2.33(0.36)**
$\langle 200, 0.7, 5\rangle$	6.47(0.23)(+)	6.46(0.28)(+)	6.48(0.29)(+)	6.43(0.41)(+)	**6.63(0.27)**
$\langle 200, 0.7, 8\rangle$	4.62(0.25)(=)	4.64(0.15)(=)	4.62(0.16)(=)	4.62(0.21)(=)	**4.69(0.16)**
$\langle 400, 0.3, 5\rangle$	10.94(0.97)(+)	11.56(0.73)(+)	11.65(0.54)(+)	11.08(0.87)(+)	**12.08(0.75)**
$\langle 400, 0.3, 8\rangle$	4.20(0.49)(+)	4.47(0.33)(=)	4.49(0.26)(=)	4.22(0.54)(+)	**4.62(0.30)**
$\langle 400, 0.7, 5\rangle$	7.76(1.07)(=)	7.62(1.09)(=)	7.54(1.10)(=)	7.63(1.28)(=)	**7.79(1.30)**
$\langle 400, 0.7, 8\rangle$	**7.22(0.22)**(=)	7.21(0.16)(=)	7.21(0.19)(=)	7.21(0.32)(=)	7.21(0.25)

Table 3 shows that AQCG achieves the highest overall mean and is significantly better than all four baselines. Its advantage over GP indicates that PC-based preselection can improve the search when the DS observations are informative. Within SGP-PS, RandomRep slightly improves over Random overall, consistent with the role of representative filtering in removing redundant observations.

However, AQCG remains better than RandomRep, indicating that reducing redundancy in the candidate pool is not sufficient. The final DS set should also cover complementary behavioral distinctions under the fixed observation budget.

QAlign highlights why this set-level criterion is needed. Although QAlign selects DSs with high individual quality, its overall performance remains close to GP, suggesting that individually useful DSs can still provide overlapping behavioral information. AQCG explicitly considers complementary pairwise alignment contributions, which makes the selected set more suitable for surrogate-based preselection. At the scenario level, AQCG has the highest mean in seven of eight scenarios. In the remaining scenario, its mean is nearly identical to the best result and shows no significant difference. Together, these results support the premise that informative DS selection should combine representative candidates with set-level complementarity.

5.2 Surrogate Quality

To diagnose surrogate guidance independently of final scheduling performance, Table 4 reports time-weighted surrogate-quality metrics under the same training-time budget. Spearman correlation measures how well the surrogate preserves the fitness ordering of estimated individuals, while top-hit rates measure whether promising individuals are correctly identified for preselection.

Table 4. Time-weighted mean (std) surrogate quality of SGP-PS variants.

	SGP-PS			
Metric	Random	RandomRep	QAlign	AQCG
Spearman	0.750(0.023)(+)	0.770(0.022)(=)	0.730(0.054)(+)	**0.780(0.034)**
Top-10% hit	0.337(0.047)(+)	0.371(0.058)(=)	0.234(0.049)(+)	**0.403(0.079)**
Top-20% hit	0.488(0.052)(+)	0.523(0.064)(=)	0.381(0.055)(+)	**0.551(0.077)**

Table 4 shows that AQCG obtains the highest mean on all three surrogate-quality metrics. Compared with Random, AQCG is significantly better on all three metrics, with the largest relative gains in the top-hit rates. This suggests that coverage-aware DS selection helps the surrogate identify promising individuals for preselection, rather than only improving global rank correlation. RandomRep narrows the gap and is not significantly different from AQCG, which is consistent with representative filtering reducing redundancy in randomly sampled DSs. However, QAlign is consistently worse than AQCG, especially on the top-hit metrics. These findings support the main performance result by showing that selecting individually high-quality DSs is less reliable than selecting a set that covers complementary pairwise alignment information.

6 Conclusion

This paper investigates informative DS selection under a fixed observation budget for phenotypic characterization in surrogate-assisted genetic programming. We propose a multi-stage framework that generates a raw DS pool, builds a representative pool through state- and behavior-based deduplication, and selects the final set using AQCG. AQCG measures how well each DS's induced behavioral distinctions align with true performance differences, then greedily constructs the final set by maximizing coverage of alignment contributions across policy pairs.

Experimental results on DCTTS demonstrate that AQCG achieves the best overall performance among the compared methods and significantly outperforms all four baselines in the overall comparison. The surrogate-quality analysis further supports this result, showing that AQCG improves the surrogate's ability to identify promising offspring compared with Random and QAlign. RandomRep remains close to AQCG, indicating that representative filtering is helpful, but AQCG's performance advantage shows that filtering alone is not sufficient. Overall, the results suggest that the final DS set should not only reduce redundant observations, but also cover complementary behavioral distinctions under the fixed observation budget.

Future work can extend this framework to other dynamic scheduling problems and explore dynamic DS selection strategies that adapt the DS set during the evolutionary process.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 72071116, and in part by the Ningbo Science and Technology Bureau under Grants 2023Z237 and 2025Z197. We are grateful to the Digital Port Technologies Lab at the University of Nottingham Ningbo China for its support, and to the university for access to its High Performance Computing Facility.

References

1. Branke, J., Nguyen, S., Pickardt, C.W., Zhang, M.: Automated design of production scheduling heuristics: A review. IEEE Transactions on Evolutionary Computation **20**(1), 110–124 (2016)
2. Burke, E.K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., Qu, R.: Hyper-heuristics: a survey of the state of the art. Journal of the Operational Research Society **64**(12), 1695–1724 (Dec 2013)
3. Chen, X., Bai, R., Qu, R., Dong, H.: Cooperative double-layer genetic programming hyper-heuristic for online container terminal truck dispatching. IEEE Transactions on Evolutionary Computation **27**(5), 1220–1234 (2023)
4. Chen, X., Bai, R., Qu, R., Dong, J., Jin, Y.: Deep reinforcement learning assisted genetic programming ensemble hyper-heuristics for dynamic scheduling of container port trucks. IEEE Transactions on Evolutionary Computation **29**(4), 1371–1385 (2025)

5. Gil-Gala, F.J., Mencía, C., Sierra, M.R., Varela, R.: Evolving priority rules for on-line scheduling of jobs on a single machine with variable capacity over time. Applied Soft Computing **85**, 105782 (2019)
6. Gil-Gala, F.J., Sierra, M.R., Mencía, C., Varela, R.: Surrogate model for memetic genetic programming with application to the one machine scheduling problem with time-varying capacity. Expert Systems with Applications **233**, 120916 (2023)
7. Guo, T., Mei, Y., Tang, K., Du, W.: Cooperative co-evolution for large-scale multiobjective air traffic flow management. IEEE Transactions on Evolutionary Computation **28**(6), 1644–1658 (2024)
8. Guo, T., Mei, Y., Zhang, M., Sun, R., Zhu, Y., Du, W.: Genetic programming with multifidelity surrogates for large-scale dynamic air traffic flow management. IEEE Transactions on Evolutionary Computation **29**(6), 2671–2685 (2025)
9. Hildebrandt, T., Branke, J.: On using surrogates with genetic programming. Evolutionary computation **23**(3), 343–367 (2015)
10. Jin, C., Bai, R., Zhou, Y., Chen, X., Tan, L.: Enhancing online yard crane scheduling through a two-stage rollout memetic genetic programming. Memetic Computing **16**(3), 467–489 (2024)
11. Jin, J., Cui, T., Bai, R., Qu, R.: Container port truck dispatching optimization using real2sim based deep reinforcement learning. European Journal of Operational Research **315**(1), 161–175 (2024)
12. Jin, J., Zhao, W., Qu, R., Ren, J., Chen, X., Zhang, Q., Bai, R.: Preference-agile multi-objective optimization for real-time vehicle dispatching. European Journal of Operational Research (2026)
13. Jin, Y.: A comprehensive survey of fitness approximation in evolutionary computation. Soft computing **9**(1), 3–12 (2005)
14. Koza, J.R.: Genetic programming: on the programming of computers by means of natural selection. MIT press (1992)
15. Langdon, W.B., Poli, R., McPhee, N.F., Koza, J.R.: Genetic Programming: An Introduction and Tutorial, with a Survey of Techniques and Applications, pp. 927–1028. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
16. Spearman, C.: The proof and measurement of association between two things. The American Journal of Psychology **15**(1), 72–101 (1904)
17. Tan, L., Jin, C., Chen, X., Qu, R., Bai, R.: PGU-SGP: A Pheno-Geno unified surrogate genetic programming for real-life container terminal truck scheduling. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 322–330. GECCO '25, Association for Computing Machinery, New York, NY, USA (2025). https://doi.org/10.1145/3712256.3726326
18. Tu, C., Bai, R., Aickelin, U., Zhang, Y., Du, H.: A deep reinforcement learning hyper-heuristic with feature fusion for online packing problems. Expert Systems with Applications **230**, 120568 (2023)
19. Xu, M., Mei, Y., Zhang, F., Zhang, M.: Genetic programming with lexicase selection for large-scale dynamic flexible job shop scheduling. IEEE Transactions on Evolutionary Computation **28**(5), 1235–1249 (2024)
20. Yang, X., Li, Y., Cui, T., Bai, R.: A collaborative evolutionary reinforcement learning approach to dynamic pickup and delivery challenges. Transportation Research Part E: Logistics and Transportation Review **212**, 104885 (2026)
21. Zhang, F., Mei, Y., Nguyen, S., Tan, K.C., Zhang, M.: Instance-rotation-based surrogate in genetic programming with brood recombination for dynamic job-shop scheduling. IEEE Transactions on Evolutionary Computation **27**(5), 1192–1206 (2023)

22. Zhang, F., Mei, Y., Nguyen, S., Zhang, M.: Phenotype based surrogate-assisted multi-objective genetic programming with brood recombination for dynamic flexible job shop scheduling. In: 2022 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1218–1225 (2022)
23. Zhang, F., Mei, Y., Nguyen, S., Zhang, M.: Survey on genetic programming and machine learning techniques for heuristic design in job shop scheduling. IEEE Transactions on Evolutionary Computation **28**(1), 147–167 (2024)
24. Zhang, Y., Bai, R., Qu, R., Tu, C., Jin, J.: A deep reinforcement learning based hyper-heuristic for combinatorial optimisation with uncertainties. European Journal of Operational Research **300**(2), 418–427 (2022)
25. Zhu, L., Zhang, F., Zhu, X., Chen, K., Zhang, M.: Sample-aware surrogate-assisted genetic programming for scheduling heuristics learning in dynamic flexible job shop scheduling. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 384–392. ACM, Lisbon, Portugal (2023)