Weighted Spearman Phenotypic Characterization for Surrogate-Assisted Genetic Programming in Dynamic Vehicle 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, 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) has become an effective approach for reducing the computational burden of automated heuristic design in dynamic scheduling, where fitness evaluation is typically dominated by repeated stochastic simulation. Its effectiveness, however, depends critically on the characterization used to represent GP individuals for surrogate modeling. Conventional top-1 phenotypic characterization (PC) retains only a top-1 signal from each decision situation, thereby discarding most of the induced preference ordering and yielding a coarse behavioral representation for distance-based surrogates. This paper proposes weighted Spearman phenotypic characterization (WSPC), which maps each individual to a fixed-length behavioral embedding by comparing the complete ranking it induces in each decision situation with that of a reference policy. WSPC introduces position-dependent weights into the Spearman correlation so that disagreements near the top of the ranking receive greater emphasis. The resulting characterization preserves richer behavioral information than conventional top-1 PC while remaining compatible with standard distance-based surrogates such as k-nearest neighbors. Experiments on dynamic container terminal truck scheduling show that, under the same training-time budget, WSPC attains the best average test performance, with significant gains over standard GP and top-1 PC. Diagnostic surrogate-quality results further support the effectiveness of WSPC for surrogate guidance.

Keywords: Genetic Programming · Surrogate Model · Phenotypic Characterization · Dynamic Scheduling

1 Introduction

Genetic Programming (GP) [13] has demonstrated strong capability in evolving effective heuristics for dynamic scheduling [4, 9]. However, its practical scalability

⋆ Corresponding author.

is fundamentally constrained in simulation-based environments, where fitness evaluation often requires repeated stochastic simulation over many instances. Since a GP run may evaluate thousands of candidate rules, the cumulative computational cost can quickly become prohibitive. To alleviate this bottleneck, surrogate-assisted GP (SAGP) incorporates predictive models to approximate expensive fitness evaluations at a substantially lower cost [7, 19, 24].

The effectiveness of SAGP depends critically on how GP individuals are represented. Most surrogate models used in this setting, including k-nearest neighbors, support vector regression, and Gaussian processes, assume inputs in the form of fixed-length numerical vectors [7, 11]. GP individuals, by contrast, are variable-sized symbolic programs whose syntactic structures and operational semantics cannot be directly processed by standard regression models. Consequently, surrogate quality is determined not only by the predictive model itself but also by the characterization used to represent GP individuals. If the resulting representation is too coarse or unstable, surrogate estimates may become less reliable, distort selection pressure, and ultimately impair search performance.

Existing characterization methods can be broadly categorized into genotypic and phenotypic approaches. Genotypic characterization (GC) extracts features directly from the representation of an individual, such as node frequencies or structural distances [14, 16, 17]. Phenotypic characterization, in contrast, captures the observable decision behavior induced by a rule on sampled decision situations (DSs) [7]. A commonly used phenotypic design in scheduling-oriented SAGP is a top-1 form of characterization (PC), which retains only a top-1 signal from each DS [7, 23]. This design is compact and efficient, but it discards most of the preference structure induced by the rule. As a consequence, two heuristics that agree on the same top-1 signal but differ substantially in the remaining ranking may become indistinguishable. Moreover, the resulting characterization is weakly comparable across DSs with different candidate-set sizes, since the same behavioral discrepancy can be expressed at different scales.

To address these limitations, this paper proposes weighted Spearman phenotypic characterization (WSPC) for SAGP in dynamic scheduling. The central idea is to represent a GP individual by the complete ranking it induces over the candidate actions in each DS and to compare this ranking against that of a fixed reference policy. The resulting per-DS agreement values are then stacked to form a fixed-length behavioral embedding. To reflect the practical importance of early decisions in online scheduling, WSPC incorporates position-dependent weights so that disagreements near the top of the ranking contribute more strongly than those near the tail. In this way, WSPC preserves richer behavioral information than PC while remaining compatible with existing distance-based surrogates.

The major contributions of this paper are as follows:

- We propose weighted Spearman phenotypic characterization (WSPC), a rank-based characterization that encodes the complete preference ordering induced by a GP rule in each decision situation and maps it to a fixed-length behavioral embedding relative to a reference policy.

- We introduce a position-aware weighting scheme that assigns greater importance to top-ranked disagreements, making the representation more sensitive to differences that are more consequential in dynamic scheduling.
- Results on dynamic container terminal truck scheduling show that, under the same training-time budget, WSPC achieves the best average scheduling performance among SGP-PS variants, and surrogate-quality diagnostics show better identification of promising individuals than top-1 PC.

2 Background and Related Work

2.1 Genetic Programming and Surrogate Modeling

Genetic Programming (GP) [13], a widely used hyper-heuristic approach [2, 22], is an evolutionary search paradigm that evolves computer programs or decision rules in a flexible symbolic representation space. In dynamic scheduling, GP individuals are typically interpreted as priority functions: given the current state and feasible actions, the rule assigns a score to each candidate and the scheduler executes the highest-priority action. Due to their interpretability and flexibility, GP-based rules have been applied successfully to a range of dynamic scheduling problems, including job shop scheduling [7, 15, 23], yard crane scheduling [9], and container terminal truck scheduling [3–5]. Related dynamic pickup-and-delivery problems have also motivated learning-based optimization approaches [21].

Despite these advantages, the practical scalability of GP in dynamic scheduling remains constrained by the cost of fitness evaluation. In simulation-based settings, each candidate rule may need to be assessed on multiple stochastic instances to obtain a reliable performance estimate. As population sizes, generation counts, and problem complexity increase, the cumulative computational burden can become prohibitive. This challenge has motivated the incorporation of surrogate models into GP. Surrogate models, also referred to as approximation models or meta-models, are computationally inexpensive predictors used to approximate expensive objective functions [6, 11, 12]. Common surrogate models include k-nearest neighbors, support vector regression, Gaussian processes, radial basis function models, and other similarity-based predictors [11, 12]. Despite their methodological differences, these models share a common assumption: the input is provided in the form of fixed-length numerical vectors [7].

This requirement makes characterization essential in SAGP because GP individuals are variable-sized symbolic structures that are not directly compatible with standard surrogate models. Before surrogate modeling can be performed, each individual must be mapped to an informative fixed-dimensional representation. The practical success of SAGP depends jointly on the predictive model and characterization. If the adopted characterization is coarse, unstable, or weakly aligned with task-specific behavior, proximity in the induced representation may fail to reflect meaningful behavioral or performance similarity, causing surrogate estimates to distort selection pressure and degrade search performance. These issues have motivated extensive research on characterization design.

2.2 Genotypic and Phenotypic Characterization

Characterization methods used in SAGP can be broadly categorized into genotypic and phenotypic approaches. Genotypic characterization (GC) encodes individuals through syntactic or structural properties of the GP representation, such as node frequencies, subtree statistics, or structural distances [14, 16, 17]. Since GC is derived directly from the program tree, it is generally stable across evaluations and can often be computed efficiently. However, for decision-making problems such as dynamic scheduling, structural similarity is not necessarily behavior-preserving: syntactically distinct rules may induce similar decisions, whereas small structural differences may produce markedly different scheduling trajectories. As a result, GC-based distances may correlate only weakly with true fitness differences, making them a less reliable basis for surrogate modeling.

Phenotypic characterization, in contrast, represents individuals through their observable behavior on decision situations [7, 20]. In scheduling applications, this behavior is reflected in how a rule scores, ranks, or selects candidate tasks in representative states. Such representations enable the surrogate to model similarity in terms of induced decisions rather than program syntax. In scheduling-oriented SAGP, a commonly used phenotypic form is the top-1 characterization (PC), which retains only a top-1 signal from each DS [7, 23]. Recent work has also combined GC and PC to enhance surrogate guidance [19, 24]. Nevertheless, PC remains the component encoding decision behavior in such hybrid designs, so a top-1 phenotypic component still carries substantial information loss.

2.3 Dynamic Container Terminal Truck Scheduling

We adopt dynamic container terminal truck scheduling (DCTTS) as the primary testbed in this study. A typical container terminal comprises the berth area, where quay cranes transfer containers between vessels and trucks, the yard area, where yard cranes manage storage and retrieval operations within yard blocks, and the entry-exit area, which connects the terminal to the external transportation network. Internal trucks move containers between berth-side and yard-side resources, thereby coupling these operational areas. DCTTS concerns the real-time assignment of internal trucks to transport tasks under dynamic task availability, stochastic service times, and uncertain travel conditions [3, 4, 10]. Since trucks coordinate the interactions among the terminal's principal resources, effective truck dispatching is critical for maintaining high resource utilization and terminal throughput, typically measured in TEU/h.

The DCTTS environment is simulated through an event-driven simulator, as illustrated in Fig. 1. During simulation, a decision situation (DS) is triggered whenever a truck becomes idle. In each DS, the dispatching rule determines the truck's next task from the feasible candidate set defined by the current system state. Let $\Omega(d)$ denote the feasible candidate-task set in DS d. A GP individual h acts as a scoring rule that assigns a priority value to each candidate task based on its feature vector $\mathbf{x}(d, t)$. In the current implementation, smaller scores indicate higher priority, and the candidate receiving the lowest score under h

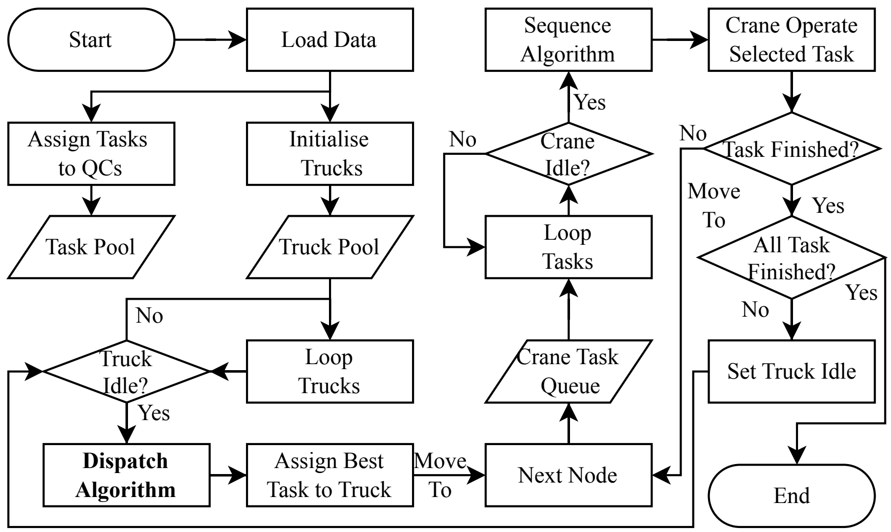

Fig. 1. Event-driven dispatch workflow of the DCTTS simulator, adapted from [19].

is selected. Repeating this procedure throughout the simulation run induces a complete ranking over the candidate tasks in each DS, which is precisely the behavioral information exploited by the proposed characterization.

3 Weighted Spearman Phenotypic Characterization

WSPC characterizes each GP individual by comparing the complete ranking it induces in each selected decision situation (DS) with that of a fixed reference policy. This reference-based design yields a fixed-length embedding over the sampled DS set while avoiding the quadratic cost of pairwise rank comparisons across individuals. In addition, since candidate-set sizes vary across DSs, it provides a more comparable basis for behavioral comparison than naive encodings based on task identities or raw rank positions.

To better reflect the importance of early decisions in dynamic scheduling, WSPC further introduces position-dependent weights into the rank comparison. As a result, disagreements near the top of the ranking receive greater emphasis, while lower-ranked tasks remain represented. This yields a richer behavioral characterization than conventional top-1 PC.

3.1 Formulation

Since rank correlation is meaningful only when at least two candidates are available, we only use DSs with $n_i \geq 2$. Singleton candidate sets are discarded because they contain no ranking information. Let $\mathcal{D} = \{d_i\}_{i=1}^{M}$ denote the selected set of M decision situations (DSs). For each $d_i \in \mathcal{D}$, let $\Omega(d_i) = [t_{i,u}]_{u=1}^{n_i}$ denote the candidate list presented by the simulator at d_i, where u is the deterministic list index and n_i is the number of candidates at d_i (which may vary across DSs). Given GP individual h and fixed reference rule h^{ref}, both assign a real-valued priority score $\pi_i(\cdot, t) \in \mathbb{R}$ to each $t \in \Omega(d_i)$. In the current study, h^{ref} is instantiated as the deterministic manual baseline [4, 19].

In our simulator, smaller priority scores indicate higher preferences. Accordingly, the rank of each candidate is obtained by sorting $\Omega(d_i)$ in ascending order of $\pi_i(\cdot, t)$, with ties broken deterministically by the index u. This yields unique ranks under both h and h^{ref}. For each $t_{i,u} \in \Omega(d_i)$, let $r_i(h, t_{i,u}) \in \{1, \ldots, n_i\}$ denote its rank under h (1 indicates the highest priority), and let $r_i(h^{\text{ref}}, t_{i,u}) \in \{1, \ldots, n_i\}$ denote the corresponding rank under h^{ref}.

To emphasize discrepancies near the top of the ranking without discarding lower-ranked tasks, we use the following monotone discount inspired by discounted cumulative gain (DCG) [8]:

$$g(k) = \frac{1}{\log_2(1+k)}, \qquad k = 1, \ldots, n_i. \tag{1}$$

We then assign each candidate a normalized weight based on its reference rank:

$$w_i(u) = \frac{g\big(r_i(h^{\text{ref}}, t_{i,u})\big)}{\sum_{j=1}^{n_i} g(r_i(h^{\text{ref}}, t_{i,j}))}, \qquad u = 1, \ldots, n_i. \tag{2}$$

By construction, $w_i(u) \geq 0$ and $\sum_{u=1}^{n_i} w_i(u) = 1$, so all candidates remain represented while higher-priority reference choices receive larger weights. For each d_i, the reference ranks $\{r_i(h^{\text{ref}}, t_{i,u})\}_{u=1}^{n_i}$ and weights $\{w_i(u)\}_{u=1}^{n_i}$ depend only on d_i and h^{ref}, and can therefore be precomputed and reused across individuals.

Using the precomputed weights $\{w_i(u)\}_{u=1}^{n_i}$, we compute the weighted Spearman correlation [18] between h and h^{ref} in each $d_i \in \mathcal{D}$ via weighted Pearson correlation on the induced rank lists. For each $t_{i,u} \in \Omega(d_i)$ $(u = 1, \ldots, n_i)$, define

$$r_{i,u}(h) = r_i(h, t_{i,u}), \qquad r_{i,u}^{\text{ref}} = r_i(h^{\text{ref}}, t_{i,u}). \tag{3}$$

The corresponding weighted means are

$$\bar{r}_i(h) = \sum_{u=1}^{n_i} w_i(u)\, r_{i,u}(h), \qquad \bar{r}_i^{\text{ref}} = \sum_{u=1}^{n_i} w_i(u)\, r_{i,u}^{\text{ref}}. \tag{4}$$

Define the weighted covariance and variances

$$\begin{aligned} S_i^{h,\text{ref}}(h) &= \sum_{u=1}^{n_i} w_i(u)\big(r_{i,u}(h) - \bar{r}_i(h)\big)\big(r_{i,u}^{\text{ref}} - \bar{r}_i^{\text{ref}}\big), \\ S_i^{h,h}(h) &= \sum_{u=1}^{n_i} w_i(u)\big(r_{i,u}(h) - \bar{r}_i(h)\big)^2, \\ S_i^{\text{ref},\text{ref}} &= \sum_{u=1}^{n_i} w_i(u)\big(r_{i,u}^{\text{ref}} - \bar{r}_i^{\text{ref}}\big)^2. \end{aligned} \tag{5}$$

The weighted Spearman correlation in d_i is then

$$\rho_i(h) = \frac{S_i^{h,\mathrm{ref}}(h)}{\sqrt{S_i^{h,h}(h)\, S_i^{\mathrm{ref},\mathrm{ref}}}}. \tag{6}$$

To obtain a normalized dissimilarity in $[0, 1]$, we map

$$p_i(h) = \frac{1 - \rho_i(h)}{2} \in [0, 1]. \tag{7}$$

Finally, the WSPC vector of h over all selected DSs is

$$\mathbf{p}(h) = \big[p_1(h), \ldots, p_M(h)\big] \in [0, 1]^M, \tag{8}$$

which serves as a scale-consistent behavioral embedding. This vector can be used directly for distance computation (e.g., Euclidean distance) in surrogate modeling and behavior-space grouping.

As a simple illustration, consider one DS with four candidates A, B, C, D and reference order $A \succ B \succ C \succ D$. Suppose two rules both rank A first. Conventional PC assigns the same top-1 characterization to both rules. If h_1 induces the same ranks $(1, 2, 3, 4)$ as the reference, then WSPC gives $p_i(h_1) = 0$. If h_2 induces ranks $(1, 4, 3, 2)$, the two rules still agree on the top candidate, but WSPC captures the remaining rank disagreement. Using the DCG weights in (2), approximately $(0.3904, 0.2463, 0.1952, 0.1681)$, gives $\rho_i(h_2) = 0.4013$ and $p_i(h_2) = 0.2994$ by (7). Thus WSPC separates rules that PC treats as identical.

3.2 Computational Complexity

For each selected $d_i \in \mathcal{D}$, we compute the reference ranks $\{r_i(h^{\mathrm{ref}}, t_{i,u})\}_{u=1}^{n_i}$ and the corresponding weights $\{w_i(u)\}_{u=1}^{n_i}$ once and reuse them across all individuals. Both this precomputation step and the ranking step required to compute WSPC for one individual are dominated by sorting the candidate list in each decision situation, yielding a complexity of $O\Big(\sum_{i=1}^{M} n_i \log n_i\Big)$. For a population of size N, the per-generation cost scales linearly with N.

4 Experimental Design

4.1 Simulation Environment and Datasets

All methods are evaluated using the DCTTS simulator described in Section 2.3. We construct eight scenarios using the same instance generator to cover varied operating conditions. Each scenario is indexed by three factors: the number of tasks per instance, which is either 200 or 400, the proportion of loading tasks, which is either 0.3 or 0.7, and the fleet size, which is either 5 or 8 trucks per quay crane. The eight scenarios are formed by all combinations of these factors, and each scenario is denoted by ⟨number of tasks, loading-task proportion, trucks per quay crane⟩. For each scenario, we generate 10 training instances and 100 testing instances, giving 80 training and 800 testing instances in total.

4.2 Fitness Evaluation

We adopt an average relative deviation (ARD)-based fitness definition with respect to a fixed reference policy, with a small bloat penalty [1, 19]:

$$F(h) = \frac{1}{|\mathcal{I}_{\text{train}}|} \sum_{m \in \mathcal{I}_{\text{train}}} \frac{\text{Obj}(m, h) - \text{Obj}(m, \text{ref})}{\text{Obj}(m, \text{ref})} - \lambda \cdot S(h), \tag{9}$$

where $\mathcal{I}_{\text{train}}$ is the training-instance set, $\text{Obj}(m, h)$ is the throughput of rule h on instance m, ref is the manual baseline, $S(h)$ is tree size, and λ is the penalty factor. Larger values of $F(h)$ indicate better performance.

4.3 Comparison Protocol

To isolate the effect of characterization design, the main experiments use the same pre-selection surrogate-assisted GP framework as the PC baseline [7], with a fixed 1NN surrogate as the base model. Thus, the compared SGP-PS variants differ only in their characterizations. We compare the following methods:

- **GP**: standard genetic programming without surrogate assistance.
- **SGP-PS(PC)**: pre-selection SAGP using conventional top-1 phenotypic characterization [7].
- **SGP-PS(SPC)**: pre-selection SAGP using unweighted full-ranking Spearman characterization.
- **SGP-PS(WSPC)**: pre-selection SAGP using weighted Spearman phenotypic characterization with DCG weighting.

To ensure a fair comparison among PC, SPC, and WSPC, we collect a pool of decision situations (DSs) from the training instances using a committee comprising the manual baseline and eight simple dispatching rules. These rules select tasks based on the shortest or longest travel time, the smallest or largest number of remaining tasks, the lowest or highest queue pressure, and the shortest or longest service time. The committee is used only for DS collection, while the manual baseline is retained as the fixed reference policy for all characterizations. We discard DSs with fewer than two feasible candidates and remove exact duplicates. From the remaining pool, we select $|\mathcal{D}| = 40$ DSs by candidate-set-size stratification, sampling two or three DSs for each size from 2 to 16 to provide balanced coverage of candidate-set sizes. The simulator-provided candidate order is retained, and the same selected DSs are used for PC, SPC, and WSPC. All methods are evaluated over 30 runs using the same set of random seeds, and pairwise differences are assessed using the Wilcoxon signed-rank test at $\alpha = 0.05$.

4.4 Parameter Settings

We adopt a fixed 18,000 s training-time budget so that all compared methods are evaluated under the same computational budget. All common GP parameters are kept fixed across scenarios, methods, and random seeds. The pre-selection brood size is 2. Following [19], the population size is 500, tournament size is 5, the number of elites is 10, the crossover/mutation/reproduction rates are 0.8/0.15/0.05, the maximum tree depth is 10, and the penalty factor is 10^{-7}.

Table 1. Mean (std) test performance ($\times 100$) over 30 runs.

Scenario	GP	SGP-PS(PC)	SGP-PS(SPC)	SGP-PS(WSPC)
Overall	6.30(0.31)(+)	6.54(0.33)(+)	6.61(0.33)(+)	**6.71(0.22)**
$\langle 200, 0.3, 5\rangle$	7.35(0.60)(+)	7.84(0.71)(+)	7.88(0.67)(+)	**8.28(0.65)**
$\langle 200, 0.3, 8\rangle$	1.88(0.38)(+)	2.21(0.42)(=)	2.21(0.46)(=)	**2.31(0.31)**
$\langle 200, 0.7, 5\rangle$	6.47(0.23)(+)	6.59(0.36)(=)	6.58(0.33)(=)	**6.60(0.31)**
$\langle 200, 0.7, 8\rangle$	4.62(0.25)(=)	**4.69(0.25)**(=)	4.65(0.22)(=)	4.66(0.20)
$\langle 400, 0.3, 5\rangle$	10.94(0.97)(+)	11.58(1.02)(+)	11.83(0.94)(+)	**12.29(0.64)**
$\langle 400, 0.3, 8\rangle$	4.20(0.49)(+)	4.47(0.35)(+)	4.46(0.37)(+)	**4.68(0.30)**
$\langle 400, 0.7, 5\rangle$	7.76(1.07)(=)	7.71(1.17)(=)	**8.01(1.20)**(=)	7.67(1.47)
$\langle 400, 0.7, 8\rangle$	7.22(0.22)(=)	7.22(0.28)(=)	**7.28(0.22)**(=)	7.22(0.25)

Note: Higher is better. Symbols compare SGP-PS(WSPC) with each non-WSPC method using the Wilcoxon signed-rank test at $\alpha = 0.05$: (+) better, (−) worse, and (=) no significant difference.

5 Results and Discussion

5.1 Scheduling Performance

Table 1 reports the overall and scenario-level test performance under the same training-time budget. All three SGP-PS variants improve the overall mean performance over standard GP, and SGP-PS(WSPC) achieves the best overall result. Its improvements over GP, SGP-PS(PC), and SGP-PS(SPC) are statistically significant under the Wilcoxon signed-rank test ($p < 0.001$, $p = 0.019$, and $p = 0.045$, respectively).

At the scenario level, SGP-PS(WSPC) obtains the highest mean performance in five of the eight scenarios. It is best in all four scenarios with a loading-task proportion of 0.3, with particularly pronounced gains in $\langle 200, 0.3, 5\rangle$ and $\langle 400, 0.3, 5\rangle$. When the loading-task proportion is 0.7, the differences among the SGP-PS variants are not statistically significant. SGP-PS(PC) has the highest mean in $\langle 200, 0.7, 8\rangle$, while SGP-PS(SPC) is best in both 400-task scenarios.

These results are consistent with the information retained by the three characterizations. PC uses only the top-ranked candidate and can miss differences in the remaining preference order. SPC preserves the complete ranking, but its uniform treatment of all rank positions may make the representation sensitive to less consequential tail-ranking differences. WSPC refines this full-ranking signal by emphasizing candidates that are highly ranked by the reference policy, which is appropriate for DCTTS because the dispatching rule executes only the highest-priority task at each decision point. This more decision-relevant representation can provide more effective surrogate guidance, especially when the rule has greater freedom to alter task choices. Loading tasks follow a prescribed within-crane order, whereas discharge tasks offer greater scheduling flexibility. The lower loading-task proportion of 0.3 therefore provides more scope for WSPC's surrogate guidance to translate into scheduling improvements, offering a plausible explanation for the more pronounced gains in these scenarios. The results suggest that the weighted full-ranking representation provides the most effective characterization among the compared SGP-PS variants.

Table 2. Time-weighted mean (std) surrogate quality of SGP-PS variants over 30 runs.

Metric	SGP-PS(PC)	SGP-PS(SPC)	SGP-PS(WSPC)
Spearman	0.7577(0.0409)(+)	**0.8088(0.0486)**(=)	**0.8088(0.0489)**
Top-10% hit	0.3933(0.0822)(+)	0.5267(0.0837)(=)	**0.5390(0.0664)**
Top-20% hit	0.5388(0.0852)(+)	0.6423(0.0786)(=)	**0.6695(0.0731)**

Note: Higher is better. Symbols follow Table 1.

5.2 Surrogate Quality

To further assess how characterization affects surrogate learning, Table 2 reports diagnostic surrogate-quality results under the same training-time budget. Each metric is averaged over the 18,000 s horizon according to the amount of training time covered by each observed value. Spearman correlation evaluates global rank preservation, while top-q% hit is the fraction of true-fitness top-q% individuals also placed in the surrogate-predicted top q%, directly reflecting pre-selection quality. SGP-PS(WSPC) significantly outperforms SGP-PS(PC) on all reported metrics. The SPC and WSPC variants achieve virtually identical Spearman correlations, indicating comparable global ranking quality, while WSPC attains the highest average top-10% and top-20% hit rates. This pattern is consistent with WSPC's focus on decision-relevant ranking differences. As pre-selection is applied repeatedly throughout evolution, selecting even one additional promising offspring for true evaluation may alter the population and subsequent search trajectory. Small improvements in top-k identification may therefore accumulate across generations, providing a plausible link between WSPC's surrogate guidance and its final scheduling performance.

6 Conclusion

This paper proposes weighted Spearman phenotypic characterization (WSPC) for surrogate-assisted genetic programming in dynamic scheduling. By replacing top-1 characterization with a full-ranking, reference-based, and position-aware behavioral embedding, WSPC preserves richer behavioral information while remaining compatible with existing distance-based surrogate models.

The DCTTS experiments show that WSPC achieves the best average scheduling performance across the eight tested scenarios under the adopted pre-selection SAGP framework. It significantly outperforms standard GP, SGP-PS(PC), and

SGP-PS(SPC). The surrogate-quality diagnostics show that WSPC significantly improves all reported metrics over PC and achieves the highest average top-k hit rates, supporting its ability to identify promising individuals during pre-selection. These results suggest that full-ranking phenotypic characterizations are more effective than conventional top-1 PC, and that position-aware weighting can further refine full-ranking representations.

Future work can evaluate WSPC with other distance-based surrogate models and other dynamic scheduling problems, and examine how alternative weighting schemes, reference-policy choice, and decision-situation sampling affect the robustness of phenotypic characterizations.

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 for support from the Digital Port Technologies Lab and access to the High Performance Computing Facility at the University of Nottingham Ningbo China.

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). https://doi.org/10.1109/TEVC.2015.2429314
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 (2013). https://doi.org/10.1057/jors.2013.71
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). https://doi.org/10.1109/TEVC.2022.3209985
4. Chen, X., Bai, R., Qu, R., Dong, H., Chen, J.: A data-driven genetic programming heuristic for real-world dynamic seaport container terminal truck dispatching. In: 2020 IEEE Congress on Evolutionary Computation (CEC). pp. 1–8. IEEE (2020). https://doi.org/10.1109/CEC48606.2020.9185659
5. 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). https://doi.org/10.1109/TEVC.2024.3381042
6. Forrester, A.I.J., Keane, A.J.: Recent advances in surrogate-based optimization. Progress in Aerospace Sciences **45**(1–3), 50–79 (2009). https://doi.org/10.1016/j.paerosci.2008.11.001
7. Hildebrandt, T., Branke, J.: On using surrogates with genetic programming. Evolutionary Computation **23**(3), 343–367 (2015). https://doi.org/10.1162/EVCO_a_00133
8. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems **20**(4), 422–446 (2002). https://doi.org/10.1145/582415.582418

9. 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). https://doi.org/10.1007/s12293-024-00424-4
10. 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). https://doi.org/10.1016/j.ejor.2023.11.038
11. Jin, Y.: A comprehensive survey of fitness approximation in evolutionary computation. Soft Computing **9**(1), 3–12 (2005). https://doi.org/10.1007/s00500-003-0328-5
12. Jin, Y.: Surrogate-assisted evolutionary computation: Recent advances and future challenges. Swarm and Evolutionary Computation **1**(2), 61–70 (2011). https://doi.org/10.1016/j.swevo.2011.05.001
13. Koza, J.R.: Genetic programming as a means for programming computers by natural selection. Statistics and Computing **4**(2), 87–112 (1994). https://doi.org/10.1007/BF00175355
14. Moraglio, A., Poli, R.: Geometric landscape of homologous crossover for syntactic trees. In: 2005 IEEE Congress on Evolutionary Computation. vol. 1, pp. 427–434. IEEE (2005). https://doi.org/10.1109/CEC.2005.1554715
15. Nguyen, S., Mei, Y., Zhang, M.: Genetic programming for production scheduling: A survey with a unified framework. Complex & Intelligent Systems **3**(1), 41–66 (2017). https://doi.org/10.1007/s40747-017-0036-x
16. Pawlik, M., Augsten, N.: Efficient computation of the tree edit distance. ACM Transactions on Database Systems **40**(1), 3:1–3:40 (2015). https://doi.org/10.1145/2699485
17. Pawlik, M., Augsten, N.: Tree edit distance: Robust and memory-efficient. Information Systems **56**, 157–173 (2016). https://doi.org/10.1016/j.is.2015.08.004
18. Spearman, C.: The proof and measurement of association between two things. The American Journal of Psychology **15**(1), 72–101 (1904). https://doi.org/10.2307/1412159
19. 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, ACM (2025). https://doi.org/10.1145/3712256.3726326
20. Vanneschi, L., Castelli, M., Silva, S.: A survey of semantic methods in genetic programming. Genetic Programming and Evolvable Machines **15**(2), 195–214 (2014). https://doi.org/10.1007/s10710-013-9210-0
21. 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). https://doi.org/10.1016/j.tre.2026.104885
22. 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). https://doi.org/10.1016/j.ejor.2021.10.032
23. 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 (2023). https://doi.org/10.1145/3583131.3590440
24. Zhu, L., Zhang, F., Zhu, X., Chen, K., Zhang, M.: Phenotype and genotype based sample aware surrogate-assisted genetic programming in dynamic flexible job shop scheduling. IEEE Transactions on Artificial Intelligence **6**(12), 3232–3247 (2025). https://doi.org/10.1109/TAI.2025.3562161