Enhancing Generalisation in Multi-objective Genetic Programming for Energy-efficient Dynamic Flexible Job Shop Scheduling

Zhuoyin Qiao✉, Fangfang Zhang, Yi Mei and Mengjie Zhang
Centre for Data Science and Artificial Intelligence & School of Engineering and Computer Science
Victoria University of Wellington
PO BOX 600, Wellington 6140, New Zealand
{zhuoyin.qiao, fangfang.zhang, yi.mei, mengjie.zhang}@ecs.vuw.ac.nz

***Abstract*—Energy-efficient dynamic flexible job shop scheduling (E-DFJSS) requires balancing production effectiveness and energy consumption. In dynamic environments, genetic programming (GP)–based hyper-heuristics are widely used to automatically learn dispatching rules. Since learned rules repeatedly guide decision-making during solution construction, their ability to perform well on unseen instances is critical. This paper proposes a generalisation-enhanced non-dominated sorting strategy for multi-objective GP in E-DFJSS, thus to learn rules with strong generalisation potential. The proposed strategy resolves dominance using purely structural statistics of evolved scheduling trees. Experimental results show improved effectiveness of non-dominated solutions on unseen scenarios. The improvements arise from higher Pareto front retention rates and larger proportions of generalist rules. Performance gains are most pronounced under high-load dynamic scheduling conditions.**

***Index Terms*—Energy-efficient job shop scheduling, Multi-objective genetic programming, Hyper-heuristic, Generalisation**

I. Introduction

Automatically designing dispatching rules for energy-efficient dynamic flexible job shop scheduling (E-DFJSS) enables effective scheduling decisions at each decision point without relying on manually crafted heuristics [1]. In E-DFJSS, dynamically arriving jobs consist of ordered operations, each assigned to a machine selected from a predefined eligible set, while machine on/off control during idle periods is commonly applied to reduce energy consumption [2]. Consequently, E-DFJSS requires the joint optimisation of three tightly coupled decision types: machine assignment (routing), operation sequencing, and machine on/off control. The strong interactions among these heterogeneous decisions significantly complicate heuristic design [3].

In real-world manufacturing environments, single-objective scheduling is often inadequate, as decision-makers must trade-off production effectiveness and energy consumption in a multi-objective setting. Production effectiveness is typically measured by indicators such as tardiness and flowtime, whereas energy performance is quantified by total energy consumption. These objectives are inherently conflicting: improving production efficiency generally requires keeping more machines active, while reducing energy consumption may degrade timeliness or workflow efficiency. Consequently, decision-makers must seek appropriate trade-offs between production effectiveness and energy consumption, a challenge that is further exacerbated in dynamic environments [4].

Genetic programming (GP) [1] has become a mainstream approach for automatically constructing scheduling rules in complex manufacturing environments such as E-DFJSS, due to its ability to evolve symbolic decision logic without manually designed templates. By adapting rule structures based on shop-floor information, GP captures complex interactions among routing, sequencing, and energy-saving decisions. Extending GP to multi-objective optimisation, multi-objective GP (MOGP) enables the exploration of trade-offs between production effectiveness and energy efficiency, producing a set of Pareto-optimal rules rather than a single heuristic. These characteristics make MOGP an effective choice for E-DFJSS.

MOGP has proven effective for automatically discovering scheduling heuristics in complex manufacturing environments [1]. However, improving the scheduling performance of learned rules on previously unseen instances remains challenging. This challenge is particularly pronounced in E-DFJSS. In such problems, routing, sequencing, and energy-saving decisions are tightly coupled, and trade-offs among conflicting objectives substantially increase the difficulty of learning generalised effective scheduling rules. To address this challenge, we propose a generalisation-enhanced non-dominated sorting strategy that incorporates structural statistics of evolved GP trees into the dominance resolution process, with the aim of improving scheduling effectiveness on unseen instances by favouring the selection of structurally robust scheduling rules. As a result, the proposed strategy provides an efficient mechanism for enhancing the scheduling performance of learned rules when applied to previously unseen instances in energy-efficient dynamic environments. The main contributions of this work are as follows:

1) A generalisation-enhanced MOGP approach is proposed for E-DFJSS, which improves the generalisation of non-dominated solutions in dynamic environments, improving the effectiveness of evolved heuristics on unseen instances.

2) Three generalisation-enhanced non-dominated sorting strategies based on rule size, terminals, and functions are developed. These strategies exploit structural information of GP individuals to guide selection toward heuristics with higher generalisation potential.
3) The experimental results demonstrate that the proposed method enhances the generalisation performance of MOGP for E-DFJSS, and it provides a promising foundation for future research on cross-environment generalisation.

II. Background

A. Problem Definition

The E-DFJSS problem involves a set of jobs $\mathfrak{J} = \{J_1, J_2, \ldots, Jn\}$, each consisting of an ordered sequence of operations $[O_{i,1}, O_{i,2}, \ldots, O_{i,o}]$ to be processed on a set of machines $\mathfrak{M} = \{M_1, M_2, \ldots, M_m\}$. Jobs arrive dynamically, and job-related information (e.g., weights w_i, due dates dd_i) becomes available upon arrival. Operations must follow their predefined order, with machine-dependent processing times. Each machine can process at most one operation at a time, operates in either an ON or OFF state [5], and cannot change state or process other operations until the current operation is completed.

To characterise both production effectiveness and energy consumption, this study considers three widely used objectives—mean flow time (Fm), mean tardiness (Tm), and mean weighted tardiness (WTm)—to evaluate production performance, and a commonly adopted objective, total energy consumption (TEC), to measure energy usage [1]. Together, these objectives define three related multi-objective problems by focusing on one of the objectives among Fm, Tm and WTm with TEC. The mathematical definitions of these objectives are given as follows:

$$Fm = \frac{1}{n}\sum_{i=1}^{n}(C_i - r_i), \tag{1}$$

$$Tm = \frac{1}{n}\sum_{i=1}^{n}\max(C_i - dd_i, 0), \tag{2}$$

$$WTm = \frac{1}{n}\sum_{i=1}^{n}W_i * \max(C_i - dd_i, 0), \tag{3}$$

$$TEC = E_{processing} + E_{idle} + E_{turning}, \tag{4}$$

where C_i denotes the completion time of job J_i, and $E_{processing}$, E_{idle} and $E_{turning}$ represent the energy consumed during processing, idle periods, and machine turn-on, respectively. By combining the above objectives, multiple E-DFJSS scenarios are constructed to evaluate the performance of the proposed algorithm under different production and energy conditions.

B. MOGP for Dynamic Scheduling

Genetic programming (GP), as a hyper-heuristic approach, evolves dispatching rules to guide scheduling decisions rather than searching the solution space directly, and has become a dominant method for dynamic scheduling problems. Non-dominated Sorting Genetic Programming (NSGP) [6] extends GP by integrating the non-dominated sorting mechanism from NSGA-II [7], and has been widely applied to multi-objective dynamic scheduling. In NSGP, each individual adopts a multi-tree representation, with each tree corresponding to a specific decision rule (e.g., routing or sequencing). At each scheduling decision point, priority values computed by these rules collectively determine the fitness of the individual. Owing to its effectiveness, the proposed algorithm is developed within the NSGP framework.

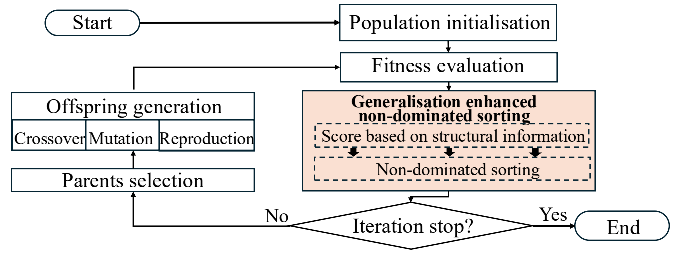

Fig. 1. The overall flowchart of the proposed method.

In this work, the MOGP evolves three trees corresponding to different scheduling decisions. Specifically, the sequencing and routing rules are priority-based, selecting the operation or machine with the highest output priority. In contrast, the energy-saving rule acts as a binary classifier: a machine is switched ON if the output is positive, and OFF otherwise.

C. Generalisation in MOGP for Dynamic Scheduling

A number of studies have investigated the generalisation ability of GP. Researchers have addressed the trade-off between model accuracy and generalisation by introducing model complexity as an additional objective [8]. In this context, a α-dominance criterion was proposed to mitigate the bias that arises when jointly considering effectiveness and model complexity. Following this, Zhang et al. [9] further incorporated a multitask learning framework to reduce this bias and improve generalisation performance. These methods follow the common principle that simpler models tend to generalise better and typically extend single-objective GP to a multi-objective formulation that balances performance and complexity. However, unlike single-objective GP, which aims to evolve a single well-generalised model, MOGP must discover a set of non-dominated heuristics that consistently generalise across multiple conflicting objectives, making the problem substantially more challenging. Research on the generalisation behaviour of MOGP has received growing attention, while its investigation in job-shop scheduling remains an open and developing topic. In this context, this paper proposes three generalisation-enhanced non-dominated sorting strategies for MOGP in E-DFJSS.

III. Generalisation-enhanced MOGP for E-DFJSS

A. Overview of the Proposed Algorithm

This study proposes a generalisation-enhanced non-dominated sorting method to improve the generalisation of non-dominated solutions evolved by GP for E-DFJSS. The overall framework is illustrated in Fig. 1. The algorithm starts with population initialisation and fitness evaluation, followed by a generalisation-enhanced non-dominated sorting procedure, where structural information of GP trees is used to compute generalisation-oriented scores that augment dominance resolution. The sorted population is then checked against the termination criterion; if unmet, parent selection and offspring generation (crossover, mutation, and reproduction) are performed. This process iterates until termination, after which the non-dominated solutions are returned. In addition, the representation encodes each individual as three rule trees, corresponding to the three decision types. During evaluation, the appropriate tree is invoked according to the current decision type. For example, at a routing decision point, the routing tree extracts shop-floor information to compute priority values for each candidate machine, and the machine with the highest priority is selected as the action for this decision point.

B. Generalisation-enhanced Non-dominated Sorting

This section describes the proposed generalisation-enhanced non-dominated sorting strategies based on structural information. Three structure-based scores are first computed for each evolved GP tree, and the subsequent non-dominated sorting is modified by using these scores to resolve dominance relations within the same Pareto level.

1) Rule Size-based Scoring: To improve the generalisation of GP individuals, a size-based non-dominated scoring strategy is introduced. Unlike prior studies that treat model size as an optimisation objective, this work focuses on the trade-off between production effectiveness and energy consumption on unseen instances. The proposed non-dominated sorting strategies incorporate model size as an auxiliary selection criterion without altering Pareto dominance, thereby favouring individuals with stronger generalisation potential. While controlling tree size helps prevent bloat, naively favouring smaller individuals can yield oversimplified structures (e.g., single-terminal trees) with insufficient expressive power for complex scheduling. Moreover, under a multi-tree representation, large size discrepancies across trees may create structural imbalance, leading to uneven representational capacity and making the learned behaviour overly sensitive to certain decision components. Therefore, the strategy simultaneously constrains the overall size of GP individuals and promotes balanced complexity across evolved trees. Based on this, the size-based score of an individual is defined as:

$$Score_{size}(i) = CD - \left((\frac{ave\overline{S} - S^i}{S^i})^2 + \frac{S^i_{max} - S^i_{min}}{ave\overline{S}_{max} - ave\overline{S}_{min}} \right), \tag{5}$$

where CD is the crowding distance [6], $ave\overline{S}$ and S^i denote the average number of nodes in the current population and the number of nodes of individual i, respectively. S^i_{max} and S^i_{min} represent the maximum and minimum tree sizes among the three trees in individual i. $ave\overline{S}_{max}$ and $ave\overline{S}_{min}$ correspond to the sizes of the tree (number of nodes) with the largest and smallest average size in the current population, respectively. To ensure a balanced contribution of different components to the overall score, both the crowding distance and size-related terms are normalised using min–max normalisation. The crowding distance is incorporated to preserve population diversity by distinguishing individuals within the same non-dominated front. The two size-related terms serve complementary purposes: one regulates the absolute complexity of individual rules, while the other enforces consistency with the population-average structural scale. Together, these terms balance individual complexity control and population-level structural alignment, thereby favouring structurally robust rules without imposing a fixed size preference. Consequently, higher scores are assigned to individuals whose complexity is close to the population-average rule size and whose constituent trees exhibit comparable structural sizes, discouraging degenerate or highly unbalanced representations while retaining sufficient expressive capacity for effective scheduling decisions.

2) Terminal-based and Function-based Scoring: Although size-based regularisation can prevent bloat and degenerate trees, size alone cannot fully capture the expressive behaviour of GP individuals. Since GP trees comprise terminals (state information) and functions (how information is combined and transformed), individuals with similar sizes may still exhibit markedly different behaviours due to differences in their functional composition. This motivates incorporating additional structure-aware criteria beyond size. Accordingly, we further exploit tree composition and design sorting strategies based on terminals and functions, respectively.

In GP, terminal symbols and function symbols play fundamentally different roles. Terminal symbols specify which state information from the shop floor is considered, whereas function symbols determine how such information is combined to generate decision priorities. Accordingly, terminal-based and function-based scoring strategies are designed to regulate different components of rule construction.

The terminal-based scoring strategy promotes information selection by evaluating the diversity and distribution of terminal symbols in each evolved tree. A very small set of unique terminals typically indicates limited state awareness, whereas entropy is used to characterise how terminals are utilised: low entropy reflects overly concentrated information usage, while excessively high entropy suggests indiscriminate utilisation that may introduce noise and reduce robustness under crossover. Notably, higher entropy is not inherently preferable, since GP can selectively emphasise informative terminals [1]; encouraging moderate terminal entropy therefore supports balanced state utilisation without undermining selective feature use.

Moreover, the function-based scoring strategy focuses on information transformation. Even with similar terminal sets, different function compositions can induce substantially different decision behaviours. Linear operators (e.g., "$+$" and "$-$") tend to induce relatively smooth priority mappings, whereas piecewise non-linear operators (e.g., "max" and "min") can introduce non-smoothness and make priority values more sensitive to local state variations, potentially leading to abrupt decision switches. Accordingly, heavy reliance on such non-smooth operators may amplify the effect of small state perturbations and reduce decision stability on unseen instances [10], [11]. Accordingly, the function-based strategy biases selection toward individuals with more stable function compositions, favouring decision rules that are more likely to generalise across dynamic environments.

Based on this, the terminal-based score and function-based score of individual i are defined as:

$$Score_{term}(i) = Score_{size}(i) + D^{norm}(i) - (\frac{\overline{H}_i}{\log K_i} - 0.5)^2, \tag{6}$$

$$\overline{H}_i = -\frac{1}{T}\sum_{t=1}^{T}\sum_{x\in U_{i,t}} \frac{c_{i,t}(x)}{\sum_{x\in U_{i,t}} c_{i,t}(x)} \log \frac{c_{i,t}(x)}{\sum_{x\in U_{i,t}} c_{i,t}(x)}, \tag{7}$$

$$Score_{func}(i) = Score_{size}(i) - \sum_{f\in F} w(f)p_i(f), \tag{8}$$

where $D^{norm}(i)$ denotes the normalised form of $D(i)$, where $D(i)$ is defined as the minimum number of unique terminals among all component trees of individual i. Specifically, $|U_{i,t}|$ represents the number of unique terminal symbols appearing in the t-th tree of individual i. The term $\overline{H}_i$ denotes the information entropy of individual i, obtained by averaging the Shannon entropy [12] computed for each evolved tree. The value $\log K_i$ corresponds to the maximum theoretical entropy, where K_i is the total number of unique terminal symbols appearing across all trees of individual i. Moreover, $c_{i,t}(x)$ represents the occurrence count of terminal x in the t-th tree of individual i. Finally, $w(f)$ and $p_i(f)$ denote the weight and the normalised frequency of function symbol f, respectively, which together characterise the contribution of different function operators in the function-based scoring strategy. In this work, with linear functions $(+,-)$ weighted as 1, functions $(*,/)$ as 2, and non-linear operators $(\max,\min)$ as 4. This design penalises structurally unstable transformations, thereby improving the potential for generalisation of the evolved heuristics.

IV. Experiment Studies

A. Simulation Model

In this study, 10 machines process 5,000 dynamically arriving jobs. Processing speeds V are randomly selected from $1, 1.3, 1.55, 1.75, 2.1$ [13]. Each machine is assigned $\varepsilon \sim U[5,10]$, with processing and idle power defined as $PP = \varepsilon V^2$ and $IP = \varepsilon/4$ [13]. Each job contains 1–10 operations; operation processing times are sampled from $U[10,99]$. Job weights $1,2,4$ are assigned with proportions 20%, 60%, and 20%, respectively. Shop-floor busyness is controlled by the utilisation level, and due dates are generated using a due-date factor of 1.2. All settings follow widely used configurations in the literature [6].

B. Parameter Setting

In the experiments, the terminal set is designed to capture shop-floor features (Table I), and all parameters follow common settings in the literature [6], [9]. The function set is $+,-,*,/,\max,\min$, where "/" is protected division returning 1 when the denominator is 0. The GP population is initialised by ramped half-and-half with size 1000. Each individual contains three trees with a maximum depth of 8 [3]. The crossover, mutation, and reproduction probabilities are 0.80, 0.15, and 0.05, respectively; terminal and non-terminal nodes are selected with probabilities 0.10 and 0.90. Tournament selection (size 7) is used for parent selection, and evolution runs for 51 generations.

TABLE I
THE GP TERMINAL AND FUNCTION SET FOR E-DFJSS.

Notation	Description
NIQ	Number of operations in the queue
WIQ	The workload in the machine queue
MWT	Waiting time that a machine becomes idle
PT	Processing time of an operation
TIS	Time in system
OWT	Waiting time of an operation in the queue
NPT	Processing time for the next operation
WKR	Work remaining for a job
NOR	The number of operations remaining for a job
MIP	Idle power of a machine
MPP	Processing power of a machine
NMPP	Processing power of next machine
SOM	Current state of a machine (ON/OFF)
NMO	Number of machines currently in the ON state
TOMO	Duration of the current machine state (ON or OFF)
TOEC	Energy consumption incurred by turning a machine ON
FMO	The frequency of state changes
Function	+, -, *, /, max, min

C. Comparison Design

To verify the proposed generalisation-enhanced sorting strategies, six scenarios are constructed from the three objectives. Scenarios with utilisation 0.75 are treated as low-load (L), while those with utilisation 0.85 are treated as high-load (H). For example, Fm-TEC_H denotes jointly optimising mean flowtime and total energy consumption in a high-load dynamic environment. For comparison, the state-of-the-art NSGP for MO-DFJSS is used as the baseline. We evaluate three proposed variants: NSGP-S (size-based), NSGP-ST (terminal-based), and NSGP-SF (function-based), and further test a weighted strategy combining terminal and function information (NSGP-STF). Each algorithm is run independently 30 times, which are evaluated on 30 unseen test instances.

To evaluate the effectiveness of the non-dominated solutions learned by different algorithms, hypervolume indicator (HV) [14] and inverted generational distance (IGD) [15] are employed. All results are analysed using the Friedman test and pairwise Wilcoxon signed-rank tests with a significance level of 0.05, where rank denotes the average ranking of each algorithm across all scenarios. The symbols "↑", "↓", and "≈" indicate that the proposed algorithm performs significantly better than, significantly worse than, or shows no significant difference from the baseline algorithm (NSGP), respectively.

D. Test Performance

TABLE II shows mean HV performance (with standard deviation) over 30 independent runs on 30 unseen instances across six E-DFJSS scenarios. Overall, NSGP-STF, which simultaneously incorporates terminal-based and function-based score information, achieves the best average rank, indicating

TABLE II

THE TEST **HV** PERFORMANCE OF PROPOSED METHODS AND THE COMPARED METHOD ACROSS SIX E-DFJSS SCENARIOS.

Scenario	NSGP	NSGP-S	NSGP-ST	NSGP-SF	NSGP-STF
Fm-TEC_H	0.89(0.08)	0.90(0.06)$^{\approx}$	0.90(0.07)$^{\approx}$	0.91(0.06)$^{\uparrow}$	0.91(0.04)$^{\uparrow}$
Tm-TEC_H	0.90(0.11)	0.94(0.06)$^{\uparrow}$	0.93(0.09)$^{\approx}$	0.93(0.07)$^{\approx}$	0.93(0.09)$^{\uparrow}$
WTm-TEC_H	0.93(0.06)	0.93(0.09)$^{\approx}$	0.90(0.11)$^{\approx}$	0.93(0.09)$^{\approx}$	0.96(0.01)$^{\uparrow}$
Fm-TEC_L	0.89(0.11)	0.92(0.09)$^{\approx}$	0.90(0.11)$^{\approx}$	0.91(0.08)$^{\approx}$	0.92(0.08)$^{\approx}$
Tm-TEC_L	0.97(0.07)	0.94(0.10)$^{\approx}$	0.97(0.08)$^{\approx}$	0.95(0.09)$^{\approx}$	0.92(0.13)$^{\approx}$
WTm-TEC_L	0.95(0.10)	0.94(0.13)$^{\approx}$	0.95(0.11)$^{\approx}$	0.98(0.05)$^{\approx}$	0.96(0.10)$^{\approx}$
W/D/L	-	1 / 5 / 0	0 / 6 / 0	1 / 5 / 0	3 / 3 / 0
Rank	3.42	3.04	2.91	2.87	2.76

TABLE III

THE TEST **IGD** PERFORMANCE OF PROPOSED METHODS AND THE COMPARED METHOD ACROSS SIX E-DFJSS SCENARIOS.

Scenario	NSGP	NSGP-S	NSGP-ST	NSGP-SF	NSGP-STF
Fm-TEC_H	0.03(0.06)	0.02(0.05)$^{\approx}$	0.02(0.04)$^{\approx}$	0.01(0.01)$^{\uparrow}$	0.01(0.04)$^{\uparrow}$
Tm-TEC_H	0.06(0.09)	0.02(0.05)$^{\uparrow}$	0.04(0.07)$^{\approx}$	0.03(0.06)$^{\approx}$	0.02(0.04)$^{\uparrow}$
WTm-TEC_H	0.04(0.08)	0.03(0.07)$^{\approx}$	0.04(0.08)$^{\approx}$	0.03(0.08)$^{\approx}$	0.01(0.01)$^{\uparrow}$
Fm-TEC_L	0.05(0.09)	0.02(0.06)$^{\approx}$	0.04(0.09)$^{\approx}$	0.03(0.06)$^{\approx}$	0.02(0.06)$^{\approx}$
Tm-TEC_L	0.02(0.06)	0.04(0.08)$^{\approx}$	0.02(0.07)$^{\approx}$	0.03(0.08)$^{\approx}$	0.06(0.11)$^{\approx}$
WTm-TEC_L	0.04(0.08)	0.05(0.11)$^{\approx}$	0.04(0.09)$^{\approx}$	0.01(0.04)$^{\approx}$	0.01(0.01)$^{\approx}$
W/D/L	-	1 / 5 / 0	0 / 6 / 0	1 / 5 / 0	3 / 3 / 0
Rank	3.32	2.95	2.98	2.96	2.79

the best test performance among all compared methods. Moreover, all proposed generalisation-enhanced variants, including NSGP-S, NSGP-ST, NSGP-SF, and NSGP-STF, achieve better average ranks than the baseline NSGP, indicating improved performance on unseen instances. Specifically, NSGP-S, NSGP-SF, and NSGP-STF significantly outperform NSGP in 1, 1, and 3 scenarios, respectively, demonstrating that incorporating terminal-level and function-level characteristics is more effective than relying on model size alone. In addition, combining these aspects of terminal and function provides further performance gains over considering each factor individually. The advantages are more pronounced under high-load scenarios, suggesting that the proposed sorting strategies are particularly effective in complex shop-floor environments where scheduling decisions are more sensitive to heuristic structure. These results highlight the importance of enhancing rule generalisation in high-load dynamic scheduling settings.

TABLE III presents the IGD performance of all compared methods over 30 independent runs on 30 unseen instances across six E-DFJSS scenarios. Overall, all proposed generalisation-enhanced variants achieve lower average IGD ranks than the baseline NSGP, indicating improved convergence and distribution of the obtained Pareto fronts on unseen instances. Among all methods, NSGP-STF attains the best average IGD rank, demonstrating its superior capability to approximate the reference Pareto front when generalising to unseen problem instances. Moreover, NSGP-SF and NSGP-STF exhibit statistically significant improvements over NSGP in multiple scenarios, particularly under high-load conditions, indicating their effectiveness in complex and congested scheduling environments. Fig. 2 further visualises the learned Pareto fronts from representative runs with medium HV for all methods across the three high-load test scenarios. Overall, the black points (NSGP-STF) consistently form the most competitive fronts, showing denser and more continuous coverage along the trade-off boundary, especially around the low-energy area. In contrast, the green points (baseline NSGP) are generally more scattered and, in tardiness-based scenarios, indicate weaker effectiveness on unseen instances.

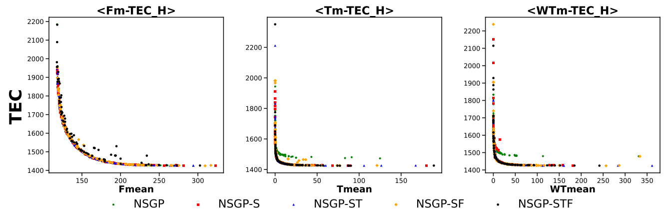

Fig. 2. Learned Pareto front of the run with medium HV values of five methods in three high-load scenarios.

Several additional insights can be drawn by jointly examining TABLEs II and III. First, the size-based strategy (NSGP-S) yields stable but limited improvements across both metrics, indicating that structural size alone provides only a weak inductive bias for generalisation in complex E-DFJSS scenarios. Finally, the superior and consistent performance of NSGP-STF across both metrics indicates that integrating terminal-level and function-level information leads to more reliable evaluation of scheduling heuristics on unseen instances, particularly in high-load shop-floor environments.

E. Evaluation of Generalisation

To verify that the proposed methods improve performance on unseen instances, we present boxplots comparing five methods on the training and test across six scenarios. To assess whether the sorting strategies enhance test performance by promoting non-dominated solutions with stronger generalisation, we adopt two indicators: the PF retention rate and the proportion of generalists. The PF retention rate is the ratio of non-dominated solutions on an unseen instance to those obtained during training [16], measuring robustness on unseen instances. The proportion of generalists is the ratio of solutions that remain non-dominated across all unseen instances to the training non-dominated set [17], capturing cross-instance consistency. Together, they provide complementary views of the generalisation behaviour of non-dominated solutions.

As shown in Fig. 3 and Fig. 4, the HV and IGD distributions of five algorithms on the training and test sets are compared across six scenarios. The baseline NSGP exhibits a clear performance degradation on the test set, characterised by lower Test-HV medians, larger dispersion, and increased Test-IGD with more extreme outliers, indicating limited generalisation on unseen environments. In contrast, the proposed variants (NSGP-S, NSGP-ST, NSGP-SF, and NSGP-STF) consistently show smaller train–test gaps, maintaining higher and more stable HV while significantly reducing both the median and variability of IGD on the test set. Among them, NSGP-STF achieves the most competitive Test-HV and Test-IGD in most scenarios, demonstrating that the proposed generalisation-enhanced sorting strategies effectively improve the generalisation potential of the learned scheduling rules, particularly under high-load conditions.

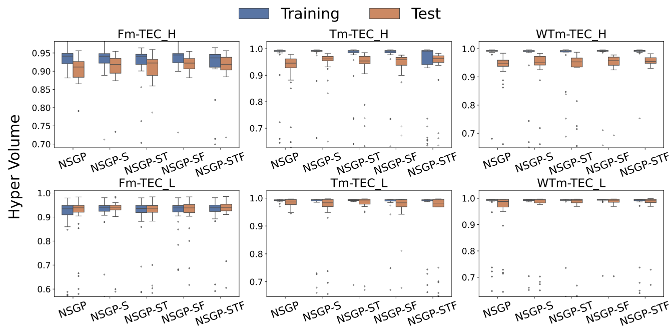

Fig. 3. Box plots of training and test **HV** values of five methods in six scenarios.

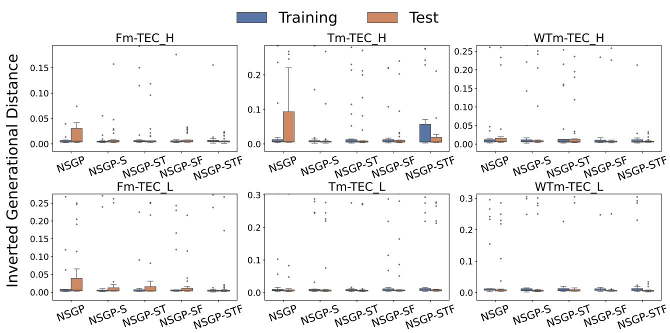

Fig. 4. Box plots of training and test **IGD** values of five methods in six scenarios.

Fig. 5 presents the corresponding heatmap visualisation. As shown in the heatmaps, the proposed sorting strategies tend to improve both the PF retention rate and the proportion of generalists compared with the baseline NSGP, with darker colours indicating higher ratios. While the rule size-based strategy (NSGP-S) already provides noticeable gains, further incorporating terminal-based and/or function-based information (NSGP-ST, NSGP-SF, and NSGP-STF) leads to additional improvements in multiple scenarios and yields better aggregated performance. NSGP-STF achieves the best average ranks on both indicators, suggesting that jointly exploiting terminal-based and function-based cues can more effectively preserve non-dominated solutions and promote the evaluation of heuristics with stronger generalisation on unseen instances.

V. Conclusions and Future Work

This paper develops a set of generalisation-enhanced sorting strategies in MOGP for E-DFJSS, aiming to guide the MOGP toward non-dominated solutions with stronger generalisation potential on unseen E-DFJSS instances. Extensive experimental studies evaluate the proposed strategies using multiple performance indicators. The results consistently show that the proposed strategies facilitate the discovery of non-dominated solutions with improved generalisation potential, thereby leading to more robust and effective scheduling performance on unseen instances. Moreover, the performance advantages are more evident under high-load scenarios. Future work will focus on further increasing the proportion of generalist non-dominated solutions and investigating cross-environment generalisation in MOGP for E-DFJSS.

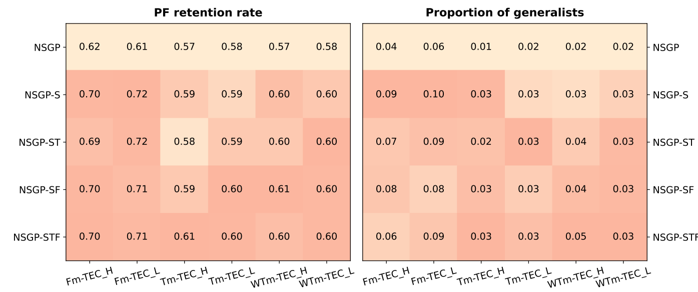

Fig. 5. Evaluating generalisation of non-dominated solutions via PF retention and generalists.

References

[1] F. Zhang, Y. Mei, S. Nguyen, and M. Zhang, “Survey on genetic programming and machine learning techniques for heuristic design in job shop scheduling,” *IEEE Trans. Evol. Comput.*, vol. 28, no. 1, pp. 147–167, 2024.

[2] G. Mouzon, M. B. Yildirim, and J. Twomey, “Operational methods for minimization of energy consumption of manufacturing equipment,” *Int. J. Prod. Res.*, vol. 45, no. 18-19, pp. 4247–4271, 2007.

[3] F. Zhang, Y. Mei, and M. Zhang, “Genetic programming with multi-tree representation for dynamic flexible job shop scheduling,” in *Australasian Joint Conference on Artificial Intelligence*, 2018, pp. 472–484.

[4] Z. Qiao, F. Zhang, Y. Mei, and M. Zhang, “Preference-based multi-objective genetic programming for energy-efficient dynamic flexible job shop scheduling,” in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2025, pp. 1–8.

[5] J. Para, J. Del Ser, and A. J. Nebro, “Energy-aware multi-objective job shop scheduling optimisation with metaheuristics in manufacturing industries: a critical survey, results, and perspectives,” *Appl. Sci.*, vol. 12, no. 3, p. 1491, 2022.

[6] F. Zhang, Y. Mei, and M. Zhang, “Evolving dispatching rules for multi-objective dynamic flexible job shop scheduling via genetic programming hyper-heuristics,” in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2019, pp. 1366–1373.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[8] G. F. Smits and M. Kotanchek, “Pareto-front exploitation in symbolic regression,” in *Genet. Program. Theory Pract. II*. Springer, 2005, pp. 283–299.

[9] F. Zhang, G. Shi, Y. Mei, and M. Zhang, “Multiobjective dynamic flexible job shop scheduling with biased objectives via multitask genetic programming,” *IEEE Trans. Artif. Intell.*, vol. 6, no. 1, pp. 169–183, 2025.

[10] E. J. Vladislavleva, G. F. Smits, and D. den Hertog, “Order of nonlinearity as a complexity measure for models generated by symbolic regression via Pareto genetic programming,” *IEEE Trans. Evol. Comput.*, 2008.

[11] M. Castelli, L. Manzoni, S. Silva, and L. Vanneschi, “A quantitative study of learning and generalization in genetic programming,” in *Proc. Eur. Conf. Genet. Program. (EuroGP)*. Springer, 2011.

[12] J. Lin, “Divergence measures based on the Shannon entropy,” *IEEE Trans. Inf. Theory*, 2002.

[13] W. Li, L. He, and Y. Cao, “Many-objective evolutionary algorithm with reference point-based fuzzy correlation entropy for energy-efficient job shop scheduling with limited workers,” *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10 721–10 734, 2022.

[14] K. Shang and H. Ishibuchi, “A new hypervolume-based evolutionary algorithm for many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 839–852, 2020.

[15] Y. Tian, X. Zhang, R. Cheng, and Y. Jin, “A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric,” in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2016, pp. 5222–5229.

[16] N. Riquelme, C. Von Lücken, and B. Baran, “Performance metrics in multi-objective optimization,” in *2015 Lat. Am. Comput. Conf. (CLEI)*. IEEE, 2015, pp. 1–11.

[17] J. Teoh, P. Varakantham, and P. Vamplew, “On generalization across environments in multi-objective reinforcement learning,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2025.