A Semantic Genetic Programming Approach to Evolving Heuristics for Multi-objective Dynamic Scheduling

Meng Xu, Yi Mei, Fangfang Zhang(✉), and Mengjie Zhang

Centre for Data Science and Artificial Intelligence & School of Engineering and Computer Science, Victoria University of Wellington, Wellington 6140, New Zealand
{meng.xu,yi.mei,fangfang.zhang,mengjie.zhang}@ecs.vuw.ac.nz

Abstract. Multi-objective dynamic flexible job shop scheduling (MO-DFJSS) is a challenging problem that requires finding high-quality schedules for jobs in a dynamic and flexible manufacturing environment, considering multiple potentially conflicting objectives simultaneously. A good approach to MO-DFJSS is to combine Genetic Programming (GP) with Non-dominated Sorting Genetic Algorithm II (NSGA-II), namely NSGP-II, to evolve a set of non-dominated scheduling heuristics. However, a limitation of NSGPII is that individuals with different genotypes can exhibit the same behaviour, resulting in a loss of population diversity. Semantic genetic programming (SGP) considers individual semantics during the evolutionary process and can enhance population diversity in various domains. However, its application in the domain of MO-DFJSS remains unexplored. Therefore, it is worthy to incorporate semantic information with NSGPII for MO-DFJSS. This study focuses on semantic diversity and semantic similarity. The results demonstrate that NSGPII considering semantic diversity yields better performance compared with the original NSGPII. Moreover, NSGPII incorporating semantic similarity achieves even better performance, highlighting the importance of maintaining a reasonable semantic distance between offspring and their parents. Further analysis reveals that the improved performance achieved by the proposed methods is attributed to the attainment of a more semantically diverse population through effective control of semantic distances between individuals.

Keywords: Heuristic learning · Multi-objective genetic programming · Semantic · Multi-objective dynamic scheduling

1 Introduction

Multi-objective dynamic flexible job shop scheduling (MO-DFJSS) is a complex combinational optimisation problem that involves scheduling multiple jobs on multiple machines in a flexible manufacturing environment, considering dynamic job arrivals and conflicting objectives. Solving the MO-DFJSS problem is challenging due to its combinatorial nature, the presence of multiple conflicting

T. Liu et al. (Eds.): AI 2023, LNAI 14472, pp. 403–415, 2024.
https://doi.org/10.1007/978-981-99-8391-9_32

objectives, and its dynamic and flexible characteristics. Scheduling heuristics [14] can quickly adapt and make real-time decisions based on the most up-to-date information, allowing for practical applications in real-world dynamic environments. However, manually designing scheduling heuristics can be particularly challenging, time-consuming, labor-intensive, and require deep domain knowledge [15]. Further, balancing multiple objectives and finding a Pareto front of high-quality scheduling heuristics require specialised algorithms and techniques, which might be difficult to incorporate into the manual design process.

Genetic programming (GP) methods have been widely used to automatically evolve scheduling heuristics for solving the DFJSS problem. There are some studies incorporating well-known Pareto dominance-based methods (i.e., non-dominated sorting genetic algorithm II [5] and strength Pareto evolutionary algorithm 2 [23]) and scalarising function-based methods (i.e., multi-objective evolutionary algorithm based on decomposition [22]) into GP, named NSGPII [19], SPGP2 [19], and MOGP/D [16], for MO-DFJSS. Among them, NSGPII showed the best performance in terms of hypervolume (HV) [24] and inverted generational distance (IGD) [7], which are two important performance indicators used to measure multi-objective algorithms [9]. However, NSGPII acts on the genotype of individuals and does not consider semantic information, which reflects the behaviour of the genotype. Semantic GP [13] has been proposed to enhance population diversity by integrating semantic information into the evolutionary process. Its effectiveness has been demonstrated across diverse domains, including symbolic regression [11], classification [1,10], and feature selection [8]. However, to the best of our knowledge, semantic information has not been incorporated into NSGPII for solving the MO-DFJSS problem. Given these promising results, it becomes particularly intriguing to explore how to improve the performance of NSGPII for MO-DFJSS by incorporating semantic information.

For this purpose, the objectives of this paper are as follows:

1. Define the *semantic* and *semantic distance* concepts in the context of MO-DFJSS domain, and design strategies to measure the semantic information derived from MO-DFJSS.
2. Design a semantic NSGPII algorithm to evolve a Pareto front of scheduling heuristics by considering the semantic information during the evolution process for solving the MO-DFJSS problem.
3. Study the effects of incorporating semantic information into NSGPII on the performance of the evolved scheduling heuristics for MO-DFJSS in terms of HV and IGD.
4. Analyse how semantic information affects the performance of evolved scheduling heuristics by NSGPII on solving the MO-DFJSS problem.

2 Background

2.1 Multi-objective Dynamic Flexible Job Shop Scheduling

In MO-DFJSS, a set of jobs $\mathcal{J} = \{J_1, J_2, ..., J_n\}$ needs to be processed by a set of machines $\mathcal{M} = \{M_1, ..., M_m\}$. Each job J_i is characterised by its arrival time r_i, due date d_i, weight w_i, and a sequence of operations $[O_{i,1}, O_{i,2}, ..., O_{i,p_i}]$ that must be performed in order. Each operation $O_{i,j}$ has a workload $\pi_{i,j}$ and can be processed by a machine from a set of optional machines $\mathcal{M}_{i,j} \subseteq \mathcal{M}$. The processing time $t_{i,j,k} = \pi_{i,j}/\gamma_k$ for operation $O_{i,j}$ on machine $M_k \in \mathcal{M}_{i,j}$ depends on the processing speed γ_k of the machine. Additionally, the machines are distributed, and there is a transport time τ_{k_1,k_2} required to move a job between two machines M_{k_1} and M_{k_2}.

The problem assumptions are as follows:

1. An operation cannot start until its preceding operation in the sequence has been completed and the job has been transported to the designated machine.
2. Each machine can handle only one operation at a time.
3. Each operation can be processed by only one of its optional machines.
4. The scheduling is non-preemptive, meaning once an operation starts, it must be completed without interruption.

This paper considers six common scheduling objectives: max-flowtime ($Fmax$), mean-flowtime ($Fmean$), max-weighted-flowtime ($WFmax$), max-tardiness ($Tmax$), max-weighted-tardiness ($WTmax$), and mean-weighted-tardiness ($WTmean$). The definitions of these objectives are as follows:

$$\begin{array}{ll} Fmax = \max_{i=1}^{n}\{C_i - r_i\}, & Fmean = \frac{1}{n}\sum_{i=1}^{n}(C_i - r_i) \\ WFmax = \max_{i=1}^{n}\{w_i(C_i - r_i)\}, & Tmax = \max_{i=1}^{n}\{T_i\} \\ WTmax = \max_{i=1}^{n}\{w_i T_i\}, & WTmean = \frac{1}{n}\sum_{i=1}^{n}(w_i T_i) \end{array}$$

where C_i is the completion time of the job J_i in the schedule, and $T_i = \max\{C_i - d_i, 0\}$ is the tardiness of the job J_i.

To facilitate analysis, we focus on bi-objective scenarios in this paper, which consider two out of the above six objectives for each scenario. More details regarding the objective selection will be provided in Sect. 4.

2.2 Related Work

MOGP for MO-DFJSS: In [19], GP is combined with two well-known Pareto dominance-based multi-objective optimisation algorithms (i.e., non-dominated sorting genetic algorithm II [5] and strength Pareto evolutionary algorithm 2 [23]) to form NSGPII and SPGP2 to evolve scheduling heuristics to address the MO-DFJSS problem. Experimental results demonstrate that NSGPII outperforms the SPGP2 in terms of both training and test HV and IGD values. Except for Pareto dominance-based methods, in [16], a multi-objective GP method based on decomposition (MOGP/D) that incorporates the advantages of multi-objective evolutionary algorithm based on decomposition [22] and GP to learn scheduling heuristics for MO-DFJSS is proposed. Among the aforementioned three classical multi-objective GP algorithms, NSGPII performs the best in terms of HV and IGD performance. Following this, some further studies were carried out on the basis of NSGPII. In [17], a novel NSGPII approach is presented for MO-DFJSS by incorporating surrogate technique and brood recombination technique. By leveraging the surrogate and brood recombination-assisted approach, the improved NSGPII obtains high-quality scheduling heuristics compared to the original NSGPII within the same training time. In [20], the influence of terminal settings on NSGPII for solving MO-DFJSS is studied. Some studies focus on interpretability [21] or multitask [18] topics in MO-DFJSS. In a word, MO-DFJSS has become a popular problem and NSGPII has become a widely used algorithm for solving it.

Semantic GP: SGP [13] has recently gained significant attention in the field of GP. It represents a valuable approach for incorporating semantic information into the evolution process, thereby improving the performance of evolved solutions. One of the key advantages of SGP is its ability to consider the behaviours/semantics rather than the genotype of individuals [13]. By considering the behaviour of individuals, SGP introduces semantic information, enabling a more nuanced understanding of the evolved solutions. The semantic analysis facilitates the discovery of individual relationships and population composition, making SGP particularly valuable in domains where different genotypes can give the same behaviour, such as MO-DFJSS.

Most SGP methods are based on the usage of genetic operators that act on the genotype to produce offspring, and then accept offspring that satisfy some semantic criteria into the next population [13]. The semantic criteria can be semantic diversity [2–4,6] and semantic similarity [11,12]. The consideration of semantic information enhances the exploration of different dimensions of search space. SGP has demonstrated its effectiveness across various problem domains, including symbolic regression [11], classification [1,10], and feature selection [8].

However, the impact of semantic information on NSGPII for MO-DFJSS has not been investigated. By investigating this, we will be able to open new avenues for solving the MO-DFJSS problem by extracting meaningful knowledge from the evolutionary process, which will be explored in this paper.

3 Methods

3.1 Overall Framework

The proposed method uses the NSGPII parent selection, crossover, and mutation to generate offspring for the next generation. On top of that, it designs novel strategies to decide which kind of offspring is allowed to be added to the next generation by considering semantic diversity and semantic similarity. In this section, we begin by providing the definitions of *semantic* and *semantic distance* in MO-DFJSS, then describe the proposed strategies. The flowchart of the improved NSGPII with the proposed strategies is shown in Fig. 1.

3.2 Semantic in MO-DFJSS

In the research domain of DFJSS, phenotypic characterisation (PC) [17] is usually used to describe the behaviour of an individual. This paper defines the

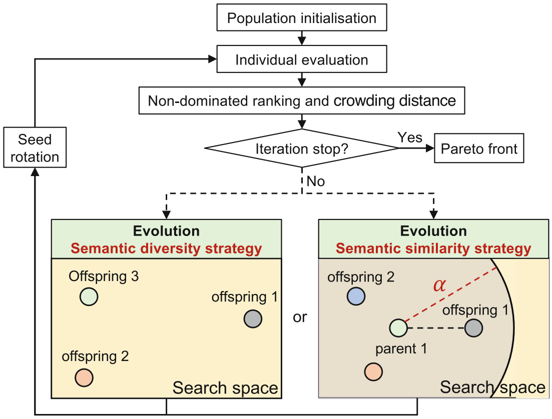

Fig. 1. The flowchart of the proposed NSGPII with the semantic diversity strategy or the semantic similarity strategy for evolution.

Table 1. An example of calculating the PC of an individual.

Sequencing				Routing			
Decision points	Reference rule	Sequencing rule	Decision	Decision points	Reference rule	Routing rule	Decision
$1(O_1)$	1	3	3	$1(M_1)$	$\underline{2}$	$\underline{1}$	2
$1(O_2)$	$\underline{3}$	$\underline{1}$		$1(M_2)$	1	2	
$1(O_3)$	2	2		$1(M_3)$	3	3	
$2(O_1)$	3	2	1	$2(M_1)$	2	2	1
$2(O_2)$	$\underline{1}$	$\underline{1}$		$2(M_2)$	3	3	
$2(O_3)$	2	3		$2(M_3)$	$\underline{1}$	$\underline{1}$	
$3(O_1)$	1	2	2	$3(M_1)$	1	2	3
$3(O_2)$	$\underline{2}$	$\underline{1}$		$3(M_2)$	$\underline{3}$	$\underline{1}$	
$3(O_3)$	3	3		$3(M_3)$	2	3	

semantic in MO-DFJSS as the PC, which is a list of decisions given by an individual on a given number of decision points. These decision points are derived by applying a reference scheduling heuristic to a given DFJSS instance. Specifically, this paper employs the weighted shortest processing time (WSPT) as the reference sequencing rule and working remaining in the queue (WIQ) as the reference routing rule. Considering that each instance often contains thousands of decision points, to save time and ease of use, we randomly select 20 sequencing decision points and 20 routing decision points, each involving a set of 7 candidates (operations for sequencing rule and machines for routing rule). Then, to calculate the PC of an individual, the sequencing/routing rule in an individual is applied to these decision points, and the ranks of the selected operations/machines across these decision points are utilised to construct the PC. An illustrative example of calculating the PC for an individual is shown in Table 1, considering 3 sequencing decision points and 3 routing decision points based on the reference scheduling heuristic. According to the given description, the PC of this example is a combination of sequencing decisions and routing decisions, which is $[3, 1, 2, 2, 1, 3]$.

Based on the PC, the *semantic distance* between individuals ind_a and ind_b is defined as the number of different decisions between their semantics and can be calculated based on Eq. (1). In contrast to other semantic methods that typically rely on Euclidean distance for calculating semantic differences, this paper proposes this definition because the Euclidean distance between machine or operation rankings in semantics is considered meaningless in DFJSS.

$$dis_{a,b} = \sum_{i=1}^{40} d_{a,b,i} \quad \text{where} \quad d_{a,b,i} = \begin{cases} 0 & \text{if } pc_{a,i} = pc_{b,i} \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

3.3 Semantic Diversity Strategy

This strategy aims to enhance the semantic diversity of the population by only accepting offspring that are semantically different from each other. Two individuals are considered semantically different if their semantic distance is greater than 0. This strategy is used whenever an offspring is generated by crossover or mutation. To be specific, when an offspring ind_o is generated, we compare it with all the other offspring already generated at the current generation. If ind_o is found to be semantically different from all the existing ones, it is accepted as a new offspring. However, if any duplicates are found, indicating that it is not semantically different, the offspring ind_o is discarded. This process is repeated iteratively until the population is filled with semantically different offspring.

3.4 Semantic Similarity Strategy

This strategy builds upon the aforementioned semantic diversity approach and introduces an additional constraint to control the *semantic similarity* between generated offspring and their parents. The degree of semantic similarity between individuals is restricted by a threshold α. Specifically, when an offspring ind_o is generated, we first compare it with all the previously generated offspring in the current generation. If ind_o is found to be semantically different from all existing ones, we further examine whether it is semantically similar to at least one of its parents. This is done by assessing whether the semantic distance between ind_o and its parent ind_p is smaller than the threshold α. If ind_o is similar to any of its parents, then it is accepted as a new offspring; otherwise, it is discarded. This process is iterated until the offspring population is filled. The idea behind this strategy is that by limiting the similarity between offspring with their parents, we expect evolution to be smooth, without losing convergence, and at the same time maintain diversity. To achieve this goal, the key point is to determine an appropriate α value.

4 Experiment Design

4.1 Dataset

This paper utilises the DFJSS simulation model [14] as an experimental tool to do the investigation. For each instance, we assume the presence of 6000 jobs (the first 1000 are warm-up jobs) that need to be processed by 10 heterogeneous machines with varying processing rates. The processing rates are randomly generated within the range of $[10, 15]$. The distances between machines and the entry/exit point are assigned using a uniform discrete distribution between 35 and 500. The transportation speed is set to 5. New jobs arrive over time according to a Poisson process. Each job consists of a random number of operations, generated from a uniform discrete distribution between 2 and 10. Jobs have different importance, represented by weights. Specifically, 20%, 60%, and 20% of jobs have weights of 1, 2, and 4, respectively. The workload for each operation is assigned using a uniform discrete distribution within the range of $[100, 1000]$. The due date for each job is determined by adding 1.5 times its processing time to its arrival time. The utilisation level plays a significant role in simulating different scenarios. A higher utilisation level indicates a busier job shop.

In this paper, six scenarios are examined by considering different combinations of objectives and different utilisation levels (0.85 and 0.95), which are: Scenario 1: $Fmax$ and $WTmax$ with 0.85; Scenario 2: $Fmax$ and $WTmax$ with 0.95; Scenario 3: $WFmax$ and $Tmax$ with 0.85; Scenario 4: $WFmax$ and $Tmax$ with 0.95; Scenario 5: $Fmean$ and $WTmean$ with 0.85; and Scenario 6: $Fmean$ and $WTmean$ with 0.95. For each scenario, 50 instances are used for training, while a separate set of 50 unseen instances is used for test.

4.2 Parameter Setting

The terminal set and function set for GP are displayed in Table 2. The terminal set comprises features associated with machines (e.g., NIQ, WIQ, and MWT), operations (e.g., PT, NPT, and OWT), jobs (e.g., WKR, NOR, rDD, SLACK, W, and TIS), and transport (e.g., TRANT). The function set consists of arithmetic operators that require two arguments. The division operator ("/") is protected and returns 1 if divided by 0. The "max" and "min" functions take two arguments and return the maximum and minimum values, respectively. Regarding the parameter configurations, the population size is set as 1000, and the Pareto front is output after 50 generations. The Ramped-half-and-half method is employed for population initialisation. The crossover and mutation rates are set to 0.85 and 0.15, respectively. Parent selection is performed using tournament selection with a size of 7.

5 Experimental Results

To simplify the algorithm description, we name NSGPII with semantic **d**iversity strategy as NSGPII^d, and NSGPII with semantic **s**imilarity strategy as NSGPII^s.

Table 2. The GP terminal and function set for DFJSS.

Notation	Description
NIQ	the number of operations in the queue
WIQ	the work in the queue
MWT	the waiting time of the machine = t* - MRT*
PT	the processing time of the operation
NPT	the median processing time for next operation
OWT	the waiting time of the operation = t - ORT*
WKR	the work remaining
NOR	the number of operations remaining
rDD	the relative due date = DD* - t
SLACK	the slack
W	the job weight
TIS	time in system = t - releaseTime*
TRANT	the transportation time
Function	$+, -, \times, /, max, min$

* t: current time; MRT: machine ready time; DD: due date; ORT: operation ready time; releaseTime: release time.

For NSGPII^s, different α values are tested, including 6, 8, 10, 12, and 14. To measure and compare the performance of algorithms, we conducted 30 independent runs for each algorithm and employed Friedman's test and Wilcoxon rank-sum test for comparison. If Friedman's test yielded significant results, we proceed with the Wilcoxon rank-sum test for pairwise comparisons between the improved NSGPII considering the semantic diversity strategy or the semantic similarity strategy and the classical NSGPII, using a significance level of 0.05.

In the subsequent results, we use the symbols "↑", "↓", and "=" to indicate statistical significance, denoting better, worse, or similar results compared to their counterparts, respectively. We use two widely used measurement indicators, HV [24] and IGD [7], to assess algorithms. A higher HV value (or a smaller IGD value) represents superior performance.

5.1 Test Performance

Tables 3 and 4 present the mean and standard deviation of the HV and IGD results for different algorithms across 30 independent runs on the test instances of the six scenarios. The bottom of the tables shows the results of the Wilcoxon comparison and Friedman's test.

For NSGPII^s, we expect evolution to be smooth, without losing convergence, and at the same time maintain diversity. To achieve this goal, the key point is to determine an appropriate α value. Therefore, we first analyse the effect of α on NSGPII^s. From the tables, we can see that NSGPII^s with $\alpha = 6$ shows significantly better HV and IGD performance across all the 6 scenarios. NSGPII^s

Table 3. The mean (standard deviation) test **HV** of 30 independent runs of NSGPII, NSGPII^d and NSGPII^s with different α for 6 scenarios.

Scenario	NSGPII	NSGPII^d	NSGPII^s				
			$\alpha = 6$	$\alpha = 8$	$\alpha = 10$	$\alpha = 12$	$\alpha = 14$
1	0.82(0.04)	0.86(0.04)	0.85(0.03)	0.86(0.04)	0.86(0.04)	0.86(0.03)	0.86(0.04)
2	0.79(0.04)	0.82(0.04)	0.84(0.03)	0.84(0.03)	0.83(0.03)	0.84(0.02)	0.82(0.03)
3	0.87(0.03)	0.88(0.03)	0.89(0.04)	0.89(0.03)	0.89(0.03)	0.89(0.03)	0.89(0.02)
4	0.95(0.01)	0.95(0.01)	0.96(0.01)	0.96(0.02)	0.96(0.01)	0.96(0.01)	0.96(0.01)
5	0.61(0.20)	0.59(0.18)	0.73(0.09)	0.66(0.12)	0.70(0.14)	0.60(0.18)	0.69(0.14)
6	0.98(0.01)	0.98(0.01)	0.98(0.01)	0.98(0.01)	0.98(0.01)	0.98(0.01)	0.98(0.01)
$\uparrow/=/\downarrow$	-	2/4/0	6/0/0	4/2/0	5/1/0	4/2/0	4/2/0
rank	6.67	5.0	2.33	3.0	3.17	4.0	3.83

Table 4. The mean (standard deviation) test **IGD** of 30 independent runs of NSGPII, NSGPII^d and NSGPII^s with different α for 6 scenarios.

Scenario	NSGPII	NSGPII^d	NSGPII^s				
			$\alpha = 6$	$\alpha = 8$	$\alpha = 10$	$\alpha = 12$	$\alpha = 14$
1	0.12(0.03)	0.10(0.03)	0.10(0.02)	0.10(0.03)	0.10(0.02)	0.09(0.02)	0.10(0.03)
2	0.12(0.03)	0.10(0.02)	0.10(0.02)	0.10(0.02)	0.10(0.02)	0.10(0.01)	0.10(0.02)
3	0.07(0.03)	0.07(0.02)	0.06(0.02)	0.06(0.02)	0.06(0.02)	0.06(0.02)	0.06(0.01)
4	0.03(0.01)	0.03(0.01)	0.02(0.01)	0.02(0.01)	0.02(0.00)	0.03(0.01)	0.02(0.01)
5	0.28(0.19)	0.29(0.17)	0.16(0.07)	0.21(0.11)	0.19(0.11)	0.28(0.18)	0.20(0.13)
6	0.01(0.01)	0.01(0.01)	0.01(0.00)	0.01(0.01)	0.01(0.01)	0.01(0.01)	0.01(0.01)
$\uparrow/=/\downarrow$	-	2/4/0	6/0/0	3/3/0	5/1/0	3/3/0	5/1/0
rank	6.5	5.17	2.67	2.67	3.17	4.17	3.67

with $\alpha = 8$, $\alpha = 12$, and $\alpha = 14$ show significantly better HV performance than NSGPII on 4 scenarios and show similar HV performance as NSGPII on the other 2 scenarios. NSGPII^s with $\alpha = 10$ shows significantly better HV performance on 5 scenarios and shows similar HV performance as NSGPII on the remaining 1 scenario. Also, NSGPII^s with $\alpha = 8$ and $\alpha = 12$ both show significantly better HV performance than NSGPII on 3 scenarios and show similar HV performance as NSGPII on the other 3 scenarios. NSGPII^s with $\alpha = 10$ and $\alpha = 14$ both show significantly better HV performance on 5 scenarios and show similar HV performance as NSGPII on the remaining 1 scenario. Based on the Friedman's test results, in terms of HV performance, NSGPII^s with $\alpha = 6$ achieves the highest rank, followed by NSGPII^s with $\alpha = 8$, NSGPII^s with $\alpha = 10$, NSGPII^s with $\alpha = 14$, NSGPII^s with $\alpha = 12$ in order. In terms of test IGD performance, both NSGPII^s with $\alpha = 6$ and $\alpha = 8$ secure the first position, NSGPII^s with $\alpha = 10$ ranks second, followed by NSGPII^s with $\alpha = 14$, NSGPII^s with $\alpha = 12$ in order. Since NSGPII^s with $\alpha = 6$ performs the best among all the NSGPII^s

with different α values, it is used for further analysis. For simplicity, we refer to NSGPII^s with $\alpha = 6$ as NSGPII^s.

Then we compare NSGPII, NSGPII^d, and NSGPII^s. We can see that, NSGPII^d shows significantly better HV and IGD performance than NSGPII on 2 scenarios and obtains similar HV and IGD performance as NSGPII on the remaining 4 scenarios. NSGPII^s gives even better performance. NSGPII^s shows significantly better HV and IGD performance than NSGPII across all the 6 scenarios. Based on the Friedman's test results, NSGPII^s ranks the first among these three methods, followed by NSGPII^d and NSGPII in order.

Through the above analysis, we can see that semantic information plays an important role in improving the performance of NSGPII on the MO-DFJSS problem. The HV and IGD performance of NSGPII can be improved by increasing the diversity of the behaviours of the individuals in the population. Moreover, requiring the offspring to have similar semantic behaviour with their parents can further improve the HV and IGD performance of NSGPII in solving the MO-DFJSS problem. This finding highlights the positive impact of considering semantic diversity and semantic similarity in NSGPII on addressing the MO-DFJSS problem.

5.2 Population Distribution

The proposed semantic diversity and semantic similarity strategies aim to limit the semantic distance between individuals in the population. It is interesting to study the semantic distribution of individuals in the population. The semantic represents the behaviour of the individual, which is a 40-dimensional vector. To visualise the semantics of individuals in the population, we employ t-SNE to reduce the dimensions to a 2-dimensional space.

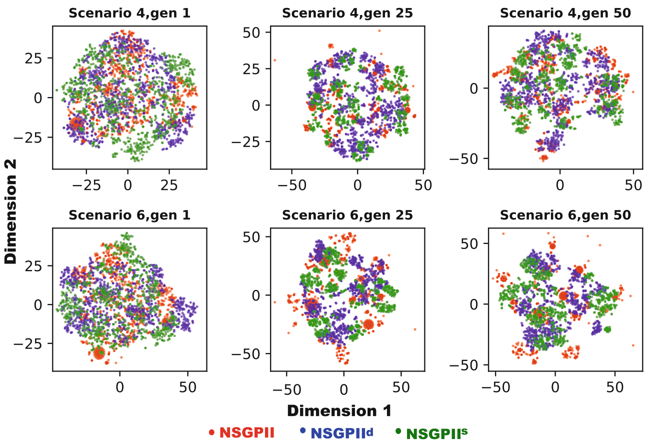

Fig. 2. Visualisations of the dimensionality reduced semantic of individuals of NSGPII, NSGPII^d, and NSGPII^s of one run in the scenario 4 and scenario 6 during the start (generation 1), middle (generation 25), and late (generation 50) stages of evolution.

Specifically, Fig. 2 visualises the dimensionality reduced semantic of individuals in the population of NSGPII, NSGPII^d, and NSGPII^s across different generations (1, 25, and 50) in scenarios 4 and 6. From Fig. 2, we can clearly see that NSGPII has several regions of more concentrated semantic distribution in each subfigure. This aligns with our expectations, as NSGPII does not impose limitations on semantic distance between individuals. Compared to NSGPII, the semantic distributions obtained by NSGPII^s, on the other hand, are relatively widespread and do not have as clearly concentrated areas as NSGPII. Compared to NSGPII and NSGPII^s, NSGPII^d gives the most diverse semantic distributions. This finding highlights the significance of restricting the semantic distance between individuals, as it allows to achieve a population with a relatively more uniform semantic distribution, avoiding losing diversity, and potentially leading to better final scheduling heuristics. These insights emphasise the importance of controlling semantic distances between individuals during the evolutionary process of NSGPII. Furthermore, it reveals that the improved final performance achieved by the inclusion of semantic information in NSGPII is attributed to its ability to evolve a more semantically diverse population.

6 Conclusions

This study has successfully demonstrated the advantages of integrating semantic information into NSGPII for addressing the MO-DFJSS problem. Firstly, this study contributes to giving the definitions of the semantic and semantic distance of scheduling heuristics for DFJSS. Then, by incorporating semantic diversity and semantic similarity within NSGPII, this study contributes to evolving better scheduling heuristics than using the original NSGPII. The results highlight the benefits of considering semantically diverse individuals for achieving high-quality scheduling heuristics. Moreover, NSGPII, considering semantic similarity, achieves the best overall performance, offering valuable insight into the importance of maintaining a reasonable semantic distance between offspring and their parents to further enhance the quality of scheduling heuristics. This emphasises the trade-off between semantic diversity and semantic similarity. Furthermore, the analysis of the population semantic distribution reveals that by controlling semantic distances between individuals, we are able to achieve a more semantically diverse population. This is the key factor contributing to the enhanced performance achieved by the proposed methods.

Overall, this paper demonstrates the potential of incorporating semantic information into the evolution process of NSGPII for MO-DFJSS, providing valuable insights into the benefits and considerations of utilising semantic information in solving complex scheduling problems. Further deeper studies are needed to explore and optimise the integration of semantic information in GP to achieve even better results for solving the MO-DFJSS problem. Some other research techniques (e.g., feature selection and surrogate) can also be combined with GP for solving the MO-DFJSS problem. In addition, although the study here is conducted on the MO-DFJSS problem, we believe that the techniques and results presented here are transferable to other complex problems. We expect the semantic information used in this work to be easily extendable to other different problems.

References

1. Bakurov, I., Castelli, M., Fontanella, F., di Freca, A.S., Vanneschi, L.: A novel binary classification approach based on geometric semantic genetic programming. Swarm Evol. Comput. **69**, 101028 (2022). https://doi.org/10.1016/j.swevo.2021.101028
2. Beadle, L., Johnson, C.G.: Semantically driven crossover in genetic programming. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 111–116 (2008)
3. Beadle, L., Johnson, C.G.: Semantic analysis of program initialisation in genetic programming. Genet. Program Evolvable Mach. **10**, 307–337 (2009)
4. Beadle, L., Johnson, C.G.: Semantically driven mutation in genetic programming. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 1336–1342 (2009)
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
6. Galván, E., Schoenauer, M.: Promoting semantic diversity in multi-objective genetic programming. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1021–1029 (2019)
7. Liu, H.L., Gu, F., Zhang, Q.: Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. IEEE Trans. Evol. Comput. **18**(3), 450–455 (2013)
8. Papa, J.P., Rosa, G.H., Papa, L.P.: A binary-constrained geometric semantic genetic programming for feature selection purposes. Pattern Recogn. Lett. **100**, 59–66 (2017)
9. Riquelme, N., Von Lücken, C., Baran, B.: Performance metrics in multi-objective optimization. In: Proceedings of the Latin American Computing Conference, pp. 1–11 (2015)
10. Sánchez, C.N., Graff, M.: Selection heuristics on semantic genetic programming for classification problems. Evol. Comput. **30**(2), 253–289 (2022)
11. Uy, N.Q., Hoai, N.X., O'Neill, M., McKay, R.I., Galván-López, E.: Semantically-based crossover in genetic programming: application to real-valued symbolic regression. Genet. Program Evolvable Mach. **12**, 91–119 (2011)
12. Uy, N.Q., McKay, B., O'Neill, M., Hoai, N.X.: Self-adapting semantic sensitivities for semantic similarity based crossover. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 1–7 (2010)
13. Vanneschi, L., Castelli, M., Silva, S.: A survey of semantic methods in genetic programming. Genet. Program Evolvable Mach. **15**(2), 195–214 (2014). https://doi.org/10.1007/s10710-013-9210-0
14. Xu, M., Mei, Y., Zhang, F., Zhang, M.: Genetic programming with cluster selection for dynamic flexible job shop scheduling. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 1–8 (2022)
15. Xu, M., Mei, Y., Zhang, F., Zhang, M.: Genetic programming with lexicase selection for large-scale dynamic flexible job shop scheduling. IEEE Trans. Evol. Comput. (2023). https://doi.org/10.1109/TEVC.2023.3244607

16. Xu, M., Mei, Y., Zhang, F., Zhang, M.: Multi-objective genetic programming based on decomposition on evolving scheduling heuristics for dynamic scheduling. In: Proceedings of the Companion Conference on Genetic and Evolutionary Computation, pp. 427–430 (2023)
17. 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: Proceedings of the IEEE Symposium Series on Computational Intelligence, pp. 1218–1225 (2022)
18. Zhang, F., Mei, Y., Nguyen, S., Zhang, M.: Multitask multiobjective genetic programming for automated scheduling heuristic learning in dynamic flexible job shop scheduling. IEEE Trans. Cybern. **53**(7), 4473–4486 (2023)
19. Zhang, F., Mei, Y., Zhang, M.: Evolving dispatching rules for multi-objective dynamic flexible job shop scheduling via genetic programming hyper-heuristics. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 1366–1373 (2019)
20. Zhang, F., Mei, Y., Zhang, M.: An investigation of terminal settings on multitask multi-objective dynamic flexible job shop scheduling with genetic programming. In: Proceedings of the Companion Conference on Genetic and Evolutionary Computation, pp. 259–262 (2023)
21. Zhang, F., Shi, G., Mei, Y.: Interpretability-aware multi-objective genetic programming for scheduling heuristics learning in dynamic flexible job shop scheduling. In: Proceedings of the IEEE Congress on Evolutionary Computation (2023)
22. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)
23. Zitzler, E., Laumanns, M., Thiele, L.: SPEA 2: improving the strength pareto evolutionary algorithm. TIK Rep. 103 (2001). https://doi.org/10.3929/ethz-a-004284029
24. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans. Evol. Comput. **7**(2), 117–132 (2003)