



HAL
open science

A New Representation and Adaptive Feature Selection for Evolving Compact Dispatching Rules for Dynamic Job Shop Scheduling with Genetic Programming

Salama Shady, Toshiya Kaihara, Nobutada Fujii, Daisuke Kokuryo

► **To cite this version:**

Salama Shady, Toshiya Kaihara, Nobutada Fujii, Daisuke Kokuryo. A New Representation and Adaptive Feature Selection for Evolving Compact Dispatching Rules for Dynamic Job Shop Scheduling with Genetic Programming. IFIP International Conference on Advances in Production Management Systems (APMS), Sep 2021, Nantes, France. pp.646-654, 10.1007/978-3-030-85906-0_70 . hal-04022134

HAL Id: hal-04022134

<https://inria.hal.science/hal-04022134>

Submitted on 9 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

A New Representation and Adaptive Feature Selection for Evolving Compact Dispatching Rules for Dynamic Job Shop Scheduling with Genetic Programming

Salama Shady*, Toshiya Kaihara, Nobutada Fujii, and Daisuke Kokuryo

Graduate School of System Informatics, Kobe University,
Kobe, Hyogo 6578501, Japan
shady.salama@kaede.cs.kobe-u.ac.jp

Abstract. Dispatching rules are extensively addressed in the dynamic job shop scheduling literature and are commonly adopted in many industrial practices. The manual design of dispatching rules is a tedious process that requires a great deal of time and experience. Due to the growth in computational power, the design process is automated using various machine learning and optimization techniques to evolve superior dispatching rules compared to human-made ones. Genetic Programming (GP) is one of the most promising approaches in the field of automated design of scheduling rules, especially under dynamic conditions. Considering a large set of terminals that reflects various job and machine attributes helps GP to obtain efficient rules, but in return extends the search space. Also, the impact of terminals can vary greatly among various scenarios, objective functions, and evolutionary stages. Therefore, an efficient feature selection mechanism can support the GP searching ability by eliminating irrelevant terminals and facilitating the process of high-quality rule search by focusing more on the promising regions in the search space. In this paper, we propose a new representation for the GP individuals that reflects the importance of each terminal in this rule. Also, an adaptive feature selection mechanism is developed that uses the information gained from the previous evolutionary step in restricting the search space at the current generation. Experimental results show that the proposed approaches assist the GP to obtain compact rules in a shorter computational time without sacrificing the performance compared with the standard GP algorithm and another representation from the literature.

Keywords: Genetic programming, job shop scheduling, feature selection.

1 Introduction

Job Shop Scheduling Problem (JSSP) is a popular combinatorial optimization problem in which a variety of custom jobs are competing to be processed in a number of general-purpose workstations in a cost-effective manner [1]. JSSPs belong to a class of problems known as NP-hard problems which means that the time required to calculate an optimal solution increases exponentially with the problem size. In Dynamic JSSP (DJSSP), jobs may arrive randomly over time with no operational information is

available before their arrival. The main sources of dynamic behavior investigated in this paper are arrival time, operational sequence, processing time, and due dates. Dispatching Rules (DR) are frequently used to generate schedules in DJSSPs because of their simplicity, ease to understand, low computational complexity, and can respond instantly to dynamic events [2]. The manual design of sophisticated DRs is a trial-and-error-cycle which requires a great deal of time, code effort and domain knowledge [2]. Therefore, several scholars have suggested the use of machine learning methods to select or generate heuristics automatically for solving combinatorial problems called "hyper-heuristics" [3]. Due to the advances in computational power and optimization approaches, there is a growing number of articles in the field of automated heuristic design [2]. Recently, Genetic Programming (GP) has been successfully applied for the design of production scheduling heuristics. There are many representations presented in the literature depending on the requirements of DRs. The tree structure is the most common representation [4]. There are other representations, such as linear expression [5], grammar-based representation, and cartesian GP [3].

Recently, the authors in [6] proposed a new GP representation where rules are defined by two parts: the priority function presented in a tree structure, and an attribute vector. The attribute vector of a rule i is a binary array $vect_i = \{x_{i1}, x_{i2}, \dots, x_{iA}\}$ with a length of A equals to the total number of terminals. If the attribute x_{ij} is 1, then this attribute is considered important (active), and the actual value of the attribute will be used, if included in the tree, in the evaluation of the priority function. In contrast, if x_{ij} is 0, the attribute is considered irrelevant (inactive), and its value will be set to 1 to exclude the effect of attribute x_j in estimating the priority value. The key limitation of this representation is that it ignores the situation where a certain attribute may not be present in the priority function. This makes the attribute vector not strictly linked to its corresponding priority function leading to the following limitations:

1. Attribute vectors do not provide sufficient, or even misleading, information regarding their priority functions. For example, all elements in an attribute vector may take the value 1 (active) even if the rule includes only one terminal.
2. Changing the activation state at a certain terminal will have no effect if that terminal is not present in this rule. However, it may have a future effect if this terminal emerges after the tree crossover or mutation, but that still is not certain.
3. The attribute vector mutation operator is applied in a uniform manner through all attributes (fixed mutation rate) without using any information gathered from previous generations and neglecting the relative importance of each terminal.

One main challenge in using the GP algorithm is that there is a large selection of terminals in the dynamic job shop that can be provided to it. Although increasing the number of features in the terminal set can provide the GP reasoning mechanism with enough information to evolve competitive rules, it exponentially expands the search space which restricts GP's ability to identify promising search regions [7]. Therefore, feature selection is a critical GP issue that may speed-up the learning process and improve the performance of evolved heuristics by selecting only informative features and eliminating irrelevant ones. To the best of our knowledge, a limited number of articles have studied feature selection to improve the GP performance in DJSSP [8].

The aim of this paper is to develop an online feature selection method using a new computationally affordable representation and an adaptive selection mechanism to estimate the weight of each feature in the terminal set using the information gained from the past generation. The rest of this paper is organized as follows. The proposed approach is described in Section 2. Section 3 presents the experimental studies in different job shop settings. The results and analysis are provided in Section 4. Finally, Section 5 gives the conclusions and future research work.

2 Proposed Feature Selection Approach

The proposed framework extends the standard GP algorithm with two main points. First, a new representation of GP individuals is proposed by modifying the attribute vector representation in [6]. Second, a feature selection approach is developed which estimates the weight of each terminal as a probability to be activated in the next generation. As shown in Figure 1, the algorithm starts by generating a population of dispatching rules using predefined sets of terminal and functions. Each rule is represented by two parts: the priority function in the standard tree structure and an attribute vector. The attribute vector proposed in this paper extends the binary representation in [6] where each terminal x_j can be active = "1" or inactive/absent="0" to a ternary array where there are three states of each terminal. If a terminal exists in the dispatching rule, it may be active = "1", or inactive = "-1", otherwise it is absent = "0". This modification leads to two main advantages over the literature representation. First, it supports the attribute vector to be precisely bound to the structure of its corresponding rule. Therefore, the attribute vectors can be used to abstract the complex structure of dispatching rules which are used in the feature selection mechanism presented later. Second, this new representation ensures that any change (mutation) of the feature's state in the attribute vector will have a direct effect on the performance of the priority function. The performance of the initial rules is assessed using a Discrete Event Simulation (DES) Model. Then, a tournament method is used for selecting the fittest rules.

Regarding the feature selection approach, the attribute vectors of a subset s of the best select individuals are used to estimate the feature activation probability. Every generation, the activation probability of each terminal x_j is estimated using Eq. (1). The Activation Probability AP_j of a given terminal j is equal to the number of times it is active divided by the sum between the active and inactive times in all chosen rules. In the most optimistic case, if a certain feature is active in all the selected rules (very important) in the current generation, its activation probability for the next generation will be 1. On the other hand, if a certain terminal is inactive in all chosen rules (irrelevant), the activation probability in the next generation will equal to 0. The activation probability serves as on-the-fly feature selection mechanism that utilizes the past evolutionary information in estimating the importance of each terminal based on its impact in the best evolved rules. The estimated activation probabilities of all the terminals are derived to the GP algorithm. Then, generic operators are applied in two steps. First, the subtree crossover and mutation are applied on the rules' priority function to create new rules for the next generation. Second, the attribute vectors are passed from

parents to offspring, then another mutation is applied to each terminal in the attribute vector. In the attribute vector mutation, the value for each attribute x_j is given using Eq. (2). For each rule i , in the absence of a certain terminal j , the value of x_{ij} in $vect_i$ will take 0. Conversely, if terminal j is presented, a uniform random number ($rand$) between 0 and 1 is generated, and two situations can occur. If $rand$ is less than or equal to its activation probability AP_j , then x_{ij} equals 1 (active), otherwise the terminal x_{ij} is inactivated and takes the value -1. Then, the new rules form the next generation. This evolutionary process is repeated until stopping criteria are met.

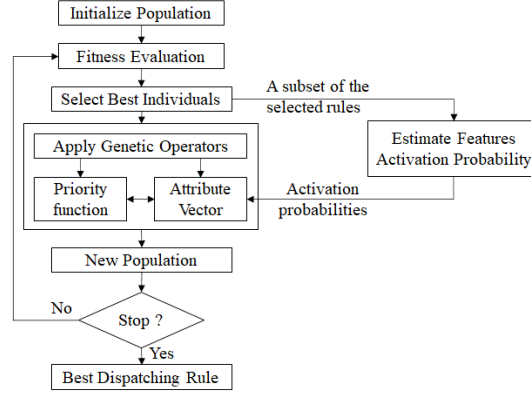


Fig. 1. The flowchart of the GP algorithm with the proposed feature selection approach.

$$AP_j = \frac{\sum_{i=1}^{|S|} (1, \text{ if } x_{ijk} = 1)}{\sum_{i=1}^{|S|} (1, \text{ if } x_{ijk} = 1) + \sum_{i=1}^{|S|} (1, \text{ if } x_{ijk} = -1)} \quad (1)$$

$$x_{ij} = \begin{cases} 0, & \text{terminal } j \text{ does not occur in rule } i \\ 1, & \text{rand} \leq AP_j \\ -1, & \text{rand} > AP_j \end{cases} \quad (2)$$

3 Experimental Design

A DES model is developed for the DJJ proposed in [9] to assess the performance of the proposed approach. Due to the importance of meeting job due dates, not only for on-time delivery jobs but also for customer satisfaction the Total Weighted Tardiness (TWT) is used as a performance measure. Also, computational time and the average length of evolved rules are considered. The job shop starts empty and jobs' arrival rate follows Poisson distribution based on the required utilization level. The total number of jobs is 2500 jobs. The statistics are collected from the 500th job (warm-up period) to the next completed 2500 jobs. The job shop settings are as follows.

- There are 10 machines in the shop floor.
- The jobs due date are assigned using the total work content [10].

- Jobs are given weights 1, 2, or 4 with probability 0.2, 0.6, and 0.2.
- The number of operations per job is uniformly distributed between 2 and 10.
- The operations processing time follows a uniform distribution between 1 and 49.

For comparison purposes, three experiments are developed to evaluate the performance of the proposed approach against the original representation in the literature [6] and the standard GP algorithm. The experiments are the Standard Genetic Programming approach (SGP), the Hybrid Genetic Programming using literature representation (HGP) [6], and the Proposed Genetic Programming with feature selection approach (PGP). As shown in Table 1, there are four utilization levels and six values for the tightness factors resulting in 6 training configurations and 24 testing scenarios. For the training phase, a single simulation replication for each configuration is used as recommended in [3]. The average normalized TWT from these scenarios is considered as the fitness for evolved rules. The WATC literature rule is used as a reference rule for the purpose of normalization as it is an efficient human-made rule in reducing TWT objective [7]. For each testing scenario, 20 independent replications are used to assess the generality of GP individuals. The terminal and function sets of the GP algorithms are given in Table 2. A population size of 1000 rules is generated using the ramped-half-and-half method with a maximum depth of 8. The crossover and mutation rates are set to 90% and 10%, respectively. The algorithm terminates after completing 80 generations. Most of these parameters are commonly used in previous studies [2]. The new parameters are the activation probability at the initial generation P_{int} and the number of selected individuals $|s|$ that are regulated to the proposed approach. Multiple pilot experiments are carried out to determine appropriate values for these parameters and evaluate their impact on the obtained results. Due to space limitations, the obtained findings are omitted and the P_{int} and $|s|$ used in the PGP experiment are 0.5 and 50, respectively. Regarding the HGP experiment, the attribute mutation probability is set to 0.5 for all defined terminals, as recommended in [6].

Table 1. The DES configurations for training and testing scenarios.

Parameter	Description
Utilization level	{0.8, 0.9} for training, and {0.8, 0.85, 0.9, 0.95} for testing
Tightness factors	{3, 6, 8} for training, and {3, 4, 5, 6, 7, 8} for testing

Table 2. Terminal and function sets.

Terminal	Description	Terminal	Description
JR	Ready time of the job	DD	Due date of the job
OR	Ready time of the operation	CT	The current time
RO	Number of operations remaining	SL	Slack of the job
WINQ	Work in the next queue	PT	Processing time
WT	Waiting time of the operation	WR	Work remaining
Npt	Next Processing time	JW	Weight of the job
Apr	Average processing time of queued jobs		
Functions	+, -, ×, /, min, max, abs, IF		

4 Results and Analysis

This section presents the obtained results for the three developed experiments. For the training phase, the three algorithms are examined using four performance indicators including computational time, total weighted tardiness, the average number of active terminals, and the average number of excluded terminals through generations. As shown in Figure 2(a), although there is no clear difference in computational requirements between SGP and HGP, the PGP achieved the lowest computational time, especially after generation 35. The reason behind this is revealed by tracking the terminals' state across generations. The gap between the average number of active features in the case of PGP and SGP widens after the 35th generation. The same trend is revealed in the number of non-existent features as depicted in Figure 2(c, d). These findings demonstrate the superiority of the proposed approach in selecting critical terminals in the evolved rules resulting in shorter rules that accelerates the evolutionary process. To verify whether reducing computational time negatively affects the performance of evolved rules. The fitness value of the best rule at each generation is tracked, as illustrated in Figure 2(b), PGP algorithm showed better convergence speed compared to the other methods with a slight difference from SGP in the last five generations, while the HGP method obtained the worst total weighted tardiness across the training scenarios. One reason may be that implementing a constant mutation probability ($P_a = 0.5$) for all terminals adversely affects the GP exploration ability. Finally, all three algorithms greatly outperform the WATC in minimizing the TWT objective.

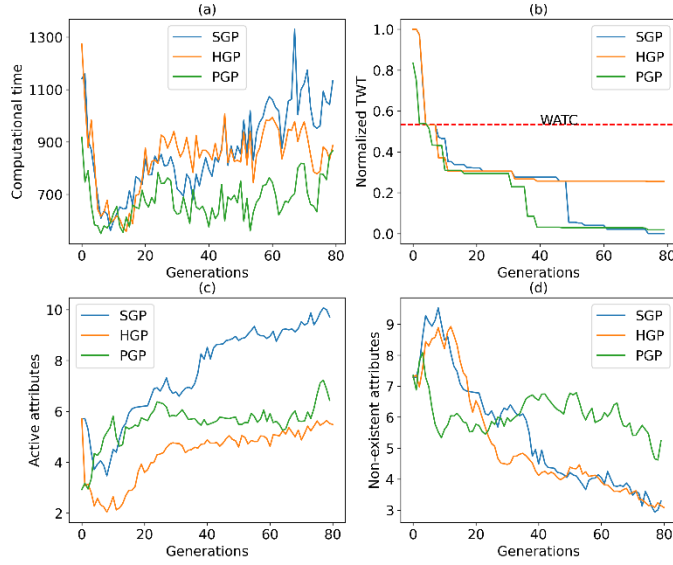


Fig. 2. The performance of the three algorithms during the training phase.

To assess the impact of the feature selection approach on the generality of the generated rules in unseen scenarios. Table 3 shows the mean (standard deviation) of the

test objective values of the HGP and PGP experiments in the 24 testing scenarios. Each scenario is denoted by a tuple (u, t) to represent a combination of job shop utilization (u) and tightness factor(t). SGP algorithm is not considered as the PGP algorithm reduced the training time by 21.8 %, therefore it is not fair to compare SGP with PGP while the stopping criterion is the maximum number of generations. Also, the Wilcoxon signed-rank test with a significance level of 0.05 is conducted between the TWT obtained by the best rules of HGP and PGP in 10 GP runs as the two methods use relatively similar representations. The statistically significant better fitness values are marked in **bold**. As shown in Table 3, PGP significantly outperforms HGP in 19/24 test scenarios while HGP performed better in three scenarios. There was no significant difference between the testing performance in two scenarios.

Moreover, the number of occurrences of each terminal in the best PGP rule is illustrated in Table 4. Two versions are shown, the ‘‘Original’’ version represents the tree structure without considering the attribute vector, and the ‘‘Modified’’ version utilizes the information given in the corresponding attribute vector to disable some terminals. The CT, PT, JW, WR, and SL terminals are the most common terminals which reflect the importance of these terminals in minimizing the TWT objective. An interesting observation is noticed, although DD terminal occurred three times in the original version, it is deactivated using the attribute vector indicating its negligible effect on the rule performance. This supports our claim that frequency analysis [7] commonly used in the literature is not an appropriate method for estimating the feature importance.

Table 3. Mean and standard deviation of total weighted tardiness on the testing scenarios

Scenario	HGP exp.	PGP exp.	Scenario	HGP exp.	PGP exp.
(0.8, 3)	68172.2(2751.0)	43382.5(1216.9)	(0.85, 5)	1734.8(95.1)	1471.8(215.7)
(0.8, 4)	3412.4(73.0)	2787.0(277.3)	(0.8, 5)	192.8(12.9)	115.9(28.4)
(0.85, 3)	216447.2(10697.1)	114657.4(1992.9)	(0.8, 6)	44.4(7.8)	32.8(5.8)
(0.85, 4)	27432.1(1549.1)	19087.1(631.5)	(0.8, 7)	19.9(3.6)	28.7(6.1)
(0.9, 3)	571280.9(17721.8)	282921.6(2157.3)	(0.8, 8)	20.5(6.1)	20.9(2.7)
(0.9, 4)	189943.5(6390.6)	98401.9(2211.5)	(0.85, 6)	117.0(9)	60.4(10.7)
(0.9, 5)	31147.6(1538.5)	20585.6(559.0)	(0.85, 7)	24.6(4.7)	31.2(2.4)
(0.95, 3)	1348638.3(30529.0)	662477.2(5800.0)	(0.85, 8)	11.3(4.1)	17.8(5.7)
(0.95, 4)	833161.7(31990.6)	397459.5(4677.0)	(0.9, 6)	2699.0(215.7)	2427.0(140.3)
(0.95, 5)	402023.4(16257.7)	196205.6(4597.5)	(0.9, 7)	206.8(16.4)	77.8(21.9)
(0.95, 6)	152141.3(5686.3)	81976.5(2588.6)	(0.9, 8)	26.4(5.1)	35.8(2.8)
(0.95, 7)	38535.0(2017.1)	21983.3(1044.7)	(0.95, 8)	5858.0(335.6)	4312.4(198.0)

Table 4. The number of times each terminal occurred in the best PGP rule.

Version	JR	OR	RO	WR	PT	DD	CT	SL	WT	Npt	WINQ	Apr	JW
Original	1	1	0	2	3	3	4	2	0	1	1	1	3
Modified	1	1	0	2	3	0	4	2	0	1	0	1	3

5 Conclusions and Future Work

In this paper, we proposed a new GP representation and an adaptive feature selection algorithm to automatically evolve compact scheduling heuristics for the DJSSP. The new representation provides an abstract view of its corresponding priority function which is easier to understand and can efficiently control the complex underlying tree structure. Moreover, the feature selection approach uses an online learning technique by utilizing the information acquired from the best rules of the previous generation in estimating the weight of each terminal. The experimental results show that the proposed approach outperforms other literature methods in terms of computational time and rule length. In future work, more detailed comparisons will be performed between the proposed approach and other feature selection methods.

References

1. Ouelhadj, D., Petrovic, S.: A survey of dynamic scheduling in manufacturing systems. *J. Sched.* 12, 417–431 (2009). <https://doi.org/10.1007/s10951-008-0090-8>.
2. Branke, J., Nguyen, S., Pickardt, C.W., Zhang, M.: Automated Design of Production Scheduling Heuristics: A Review. *IEEE Trans. Evol. Comput.* 20, 110–124 (2016). <https://doi.org/10.1109/TEVC.2015.2429314>.
3. Nguyen, S., Mei, Y., Zhang, M.: Genetic programming for production scheduling: a survey with a unified framework. *Complex Intell. Syst.* 3, 41–66 (2017). <https://doi.org/10.1007/s40747-017-0036-x>.
4. Shady, S., Kaihara, T., Fujii, N., Kokuryo, D.: A Hyper-Heuristic Framework using GP for Dynamic Job Shop Scheduling Problem. In: Proceedings of the 64th Annual Conference of the Institute of Systems, Control and Information Engineers (ISCIE). pp. 248–252 (2020).
5. Nie, L., Gao, L., Li, P., Zhang, L.: Application of gene expression programming on dynamic job shop scheduling problem. In: Proceedings of the 2011 15th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2011. pp. 291–295 (2011). <https://doi.org/10.1109/CSCWD.2011.5960088>.
6. Nguyen, S., Mei, Y., Xue, B., Zhang, M.: A hybrid genetic programming algorithm for automated design of dispatching rules. *Evol. Comput.* 27, 467–596 (2018). https://doi.org/10.1162/evco_a_00230.
7. Mei, Y., Nguyen, S., Xue, B., Zhang, M.: An Efficient Feature Selection Algorithm for Evolving Job Shop Scheduling Rules With Genetic Programming. *IEEE Trans. Emerg. Top. Comput. Intell.* 1, 339–353 (2017). <https://doi.org/10.1109/tetci.2017.2743758>.
8. Zhang, F., Mei, Y., Zhang, M.: A two-stage genetic programming hyper-heuristic approach with feature selection for dynamic flexible job shop scheduling. In: Proceedings of the 2019 Genetic and Evolutionary Computation Conference. pp. 347–355. Association for Computing Machinery, USA (2019). <https://doi.org/10.1145/3321707.3321790>.
9. Hunt, R., Richard, J., Zhang, M.: Evolving Dispatching Rules with Greater Understandability for Dynamic Job Shop Scheduling. School of Engineering and Computer Science, Victoria University of Wellington, 2016. (2016).
10. Shady, S., Kaihara, T., Fujii, N., Kokuryo, D.: Automatic Design of Dispatching Rules with Genetic Programming for Dynamic Job Shop Scheduling. In: IFIP International Conference on Advances in Production Management Systems. pp. 399–407. Springer (2020). https://doi.org/10.1007/978-3-030-57993-7_45.