# A Single Population Genetic Programming based Ensemble Learning Approach to Job Shop Scheduling

John Park[1], Su Nguyen[1,2], Mengjie Zhang[1], Mark Johnston[1]
[1]Evolutionary Computation Research Group, Victoria University of Wellington, PO Box 600, Wellington, New Zealand
[2]International University - VNU HCMC, Vietnam
{John.Park,Su.Nguyen,Mengjie.Zhang}@ecs.vuw.ac.nz, Mark.Johnston@msor.vuw.ac.nz

## ABSTRACT

Genetic Programming based hyper-heuristics (GP-HH) for dynamic job shop scheduling (JSS) problems are approaches which aim to address the issue where heuristics are only effective for specific JSS problem domains, and that designing effective heuristics for JSS problems can be difficult. This paper is a preliminary investigation into improving the robustness of heuristics evolved by GP-HH by evolving ensembles of dispatching rules from a single population of GP individuals. The results show that the current approach does not evolve significantly better or robust rules than a standard GP-HH approach of evolving single constituent rules.

## CCS Concepts

•**Computing methodologies** → **Heuristic function construction;** Ensemble methods; •**Applied computing** → *Operations research;*

## Keywords

Time-tabling and scheduling, Genetic programming, Heuristics, Combinatorial optimization, Robustness of solutions

## 1. INTRODUCTION

Job shop scheduling (JSS) problems are a set of mathematical optimisation problems which have been studied extensively due to its computational challenge and applicability to various real world scenarios [1]. This paper focuses on a dynamic JSS problem, where unforseen jobs with a sequence of operations arrive at a shop floor to be processed by the machines on the shop floor. The difficulty lies in the fact that machines can only process one job at a time, meaning that complex decisions need to be made on which jobs to process. The objective is to minimise the total weighted tardiness (TWT) [1] of the processed jobs, which is the weighted positive difference between the completion time of the jobs, i.e., the times the final operations of jobs finish processing, and their due dates.

The most prominent approach to dynamic JSS problems are dispatching rules (DRs). DRs are heuristics which select the next job to be processed by a machine currently not processing a job. Different DRs have been shown to handle specific dynamic JSS problems very well [1], but are effective only for particular problem domains. In addition, designing an effective DR for a dynamic JSS problem domain is difficult [2], requiring human experts.

This paper focuses on Genetic Programming based hyper-heuristics (GP-HH) [3]. Hyper-heuristics are heuristics, which generate heuristics that can then be applied to the problem. In the case with dynamic JSS problems, tree-based GP-HH approaches are very intuitive because individuals in the GP population can be used as a DR, where the individuals represent priority function trees that assign priorities to jobs waiting at machines. However, GP-HH evolved rules, similar to human designed DRs, suffer from the fact that DRs are myopic [4]. In addition, a single rule may not be sufficient to handle the complex decisions that need to be made in dynamic JSS problem instances.

Ensemble learning [5] and multi-agent systems [6] have been proposed in the literature to handle difficult problems. In ensemble learning, a rule is comprised of multiple smaller constituent rules, and a consensus is reached between the constituent rules to form an output. In the literature, ensemble learning have been applied to difficult classification problems [7] due to its ability to handle complex decisions.

The goal of this paper is to investigate an approach of evolving an ensemble of DRs using GP-HH for the dynamic JSS problem with the objective of minimising TWT. Furthermore, the approach will focus on developing an ensemble which is evolved in a "team" [6], and where only a single subpopulation of individuals will be used in the GP population instead of partitioning the population into multiple subpopulations, i.e., into "islands" [6].

## 2. THE METHOD

The proposed GP-HH approach for evolving ensembles of DRs will use a discrete-event simulation for evaluation. In a simulation, jobs are generated randomly until a certain number of jobs have been processed by the machines. After all generated jobs have been processed, the TWT is calculated and used as part of the rule's performance over the 'training set'. For this paper, to evolve an ensemble three distinct steps are carried out:

1. A tree-based GP is used, where the individuals represent arithmetic function trees that assign priorities to the jobs waiting at a machine in a non-delay schedule. The terminals used in the GP process are similar to the terminals used by Nguyen et al. [2], which are the basic properties of the job for which a priority value is calculated for, along with machine properties.


*GECCO '15 July 11-15, 2015, Madrid, Spain*

ACM ISBN 978-1-4503-3488-4/15/07.
DOI: http://dx.doi.org/10.1145/2739482.2764651

Table 1: Results (abridged) of the averages and standard deviations of TWTs over the 30 evolved DRs and EDRs over the training sets 4op and 8op

| Training Set | Test Set | DR | EDR |
|---|---|---|---|
| 4op | $\langle 25, 0.90\rangle$ | $44822 \pm 13334$ | $44902 \pm 13362$ |
| | $\langle 25, 0.97\rangle$ | $44536 \pm 12890$ | $44571 \pm 12884$ |
| | $\langle 50, 0.90\rangle$ | $90437 \pm 27257$ | $90661 \pm 27391$ |
| | $\langle 50, 0.97\rangle$ | $89537 \pm 26339$ | $89674 \pm 26431$ |
| 8op | $\langle 25, 0.90\rangle$ | $44635 \pm 13268$ | $44644 \pm 13249$ |
| | $\langle 25, 0.97\rangle$ | $44422 \pm 12856$ | $44421 \pm 12837$ |
| | $\langle 50, 0.90\rangle$ | $90210 \pm 27179$ | $90205 \pm 27184$ |
| | $\langle 50, 0.97\rangle$ | $89349 \pm 26312$ | $89343 \pm 26287$ |

2. The individuals in the GP population are grouped with other individuals in the population to form the ensembles. This grouping is done randomly, and multiple groupings are made for a single individual to find the overall 'performance' of the individual when it is part of an ensemble.

3. When a machine is available, the constituent rules that make up an ensemble 'vote' on the jobs that are waiting at the machine. A priority-based voting scheme is carried out, where the job with the highest sum of normalised priority assigned to it by the rules that have voted on the job is selected to be processed.

4. The evaluation procedure for an individual combines the performance of the ensembles that an individual was grouped into along with a diversity measure to generate the final fitness of the individual. The diversity measure is a modification of negative correlation learning (NCL) [8], and adapted for dynamic JSS. It compares the normalised priorities assigned to the jobs selected by machines from the individuals, which make up the ensemble, to each other.

After the GP evolutionary process, the final output is the best performing ensemble over the simulation that has been constructed during the grouping scheme of individuals.

# 3. EXPERIMENT DESIGN AND RESULTS

The GP parameters used to evolve the ensembles is as follows. The population and the number of generations is set at 256 and 25 respectively. The crossover, mutation and reproduction rates are set at 80%, 10%, and 10% respectively. Tournament selection of size 7 is used. Finally, the number of GP evolved rules is set to 30. The simulator used is the one proposed by Hunt et al. [9], where there are two training sets which generate jobs with 4 and 8 operations respectively. The training set with 4 operations per job is denoted as $4op$, and the training set with 8 operations per job is denoted as $8op$. The evolved ensemble of DRs (EDRs) are compared against standard single rule DRs evolved using the same terminals and GP parameters on 20 test simulations divided into 4 subsets based on the mean processing time ($\mu$) and utilisation rate ($\rho$) [9], which will be denoted in Table 1 as $\langle \mu, \rho \rangle$

The abridged results in Table 1 show that the current approach to evolving ensembles generate ensembles of similar performance to the single rules. A possible reason why the evolved DRs and EDRs have similar performance is that the individuals in the ensembles may not be sufficiently covering for each other's errors, resulting in the ensembles performing no better than single priority rules. Therefore, further work will be required to improve the quality of the ensembles generated.

# 4. CONCLUSIONS AND FUTURE WORKS

This initial work in progress has found that the proposed approach to evolving EDRs using GP for dynamic JSS problems generates EDRs with similar performance to single rules. Therefore, further investigation is required to develop an effective ensemble that is evolved from a single GP population of individuals. We assume that minor modifications to the grouping scheme, such as the grouping scheme proposed by Wu and Banzhaf [10], can potentially improve the quality of the evolved ensembles. In addition, the performance of the ensemble over the training set and the diversity measure can be separated into two different objectives in multi-objective fitness evaluation to potentially improve the fitness evaluation of individuals.

# 5. REFERENCES


[1] Pinedo, M.L.: Scheduling: Theory, Algorithms, and Systems. Springer (2012) 3rd edition.

[2] Nguyen, S., Zhang, M., Johnston, M., Tan, K.C.: A computational study of representations in genetic programming to evolve dispatching rules for the job shop scheduling problem. IEEE Transactions on Evolutionary Computation **17**(5) (2013) 621–639

[3] Burke, E.K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Ozcan, E., Qu, R.: Hyper-heuristics: a survey of the state of the art. Journal of the Operational Research Society **64**(12) (2013) 1695–1724

[4] Branke, J., Pickardt, C.W.: Evolutionary search for difficult problem instances to support the design of job shop dispatching rules. European Journal of Operational Research **212**(1) (2011) 22–32

[5] Polikar, R.: Ensemble based systems in decision making. IEEE Circuits and Systems Magazine **6**(3) (2006) 21–45

[6] Panait, L., Luke, S.: Cooperative multi-agent learning: The state of the art. Autonomous Agents and Multi-Agent Systems **11**(3) (2005) 387–434

[7] Bhowan, U., Johnston, M., Zhang, M., Yao, X.: Evolving diverse ensembles using genetic programming for classification with unbalanced data. IEEE Transactions on Evolutionary Computation **17**(3) (2013) 368–386

[8] Liu, Y., Yao, X.: Negatively correlated neural networks can produce best ensembles. Australian Journal of Intelligent Information Processing Systems **4**(3–4) (1997) 176–185

[9] Hunt, R., Johnston, M., Zhang, M.: Evolving "less-myopic" scheduling rules for dynamic job shop scheduling with genetic programming. In: Proceedings of the 2014 Conference on Genetic and Evolutionary Computation. GECCO 2014 (2014) 927–934

[10] Wu, S.X., Banzhaf, W.: Rethinking multilevel selection in genetic programming. In: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation. GECCO 2011 (2011) 1403–1410