Niching Genetic Programming based Hyper-heuristic Approach to Dynamic Job Shop Scheduling: An Investigation into Distance Metrics

John Park, Yi Mei, Gang Chen, Mengjie Zhang
Evolutionary Computation Research Group, Victoria University of Wellington, PO Box 600, Wellington, New Zealand
{John.Park,Yi.Mei,Aaron.Chen,Mengjie.Zhang}@ecs.vuw.ac.nz

ABSTRACT

This paper investigates the application of fitness sharing to a coevolutionary genetic programming based hyper-heuristic (GP-HH) approach to a dynamic job shop scheduling (DJSS) problem that evolves an ensemble of dispatching rules. Evolving ensembles using GP-HH for DJSS problem is a relatively unexplored area, and has been shown to outperform standard GP-HH procedures that evolve single rules. As a fitness sharing algorithm has not been applied to the specific GP-HH approach, we investigate four different phenotypic distance measures as part of a fitness sharing algorithm. The fitness sharing algorithm may potentially improve the diversity of the constituent members of the ensemble and improve the quality of the ensembles. The results show that the niched coevolutionary GP approaches evolve smaller sized rules than the base coevolutionary GP approaches, but have similar performances.

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 combinatorial optimisation problems which have been studied extensively due to their computational complexity and their applicability to various real-world manufacturing environments. This paper focuses on a dynamic JSS (DJSS) problem with unforeseen job arrivals, where a job's arrival and its properties are unknown until it reaches the shop floor. An arriving job has a sequence of *operations* that need to be completed, where a specific machine is required to process a job's operation. The objective is to complete all arriving jobs and minimise the total weighted tardiness (TWT) from processing the jobs. A job j's tardiness T_j is the positive difference between the completion time C_j of the job and the due date d_j, i.e., $T_j = \max\{C_j - d_j, 0\}$, and the weighted tardiness of job j is the weight w_j of the job times its tardiness T_j.

One technique of handling DJSS problems is to evolve *reusable* heuristics to solve DJSS problem instances. Therefore, many effective genetic programming based hyper-heuristic (GP-HH) approaches have been proposed for DJSS problems [1] that automatically evolve *dispatching rules*. A dispatching rule is a low-level heuristic that determines the next job to be processed by a machine when it becomes available at a *dispatching decision*. This bypasses the need for human experts and lengthy trial-and-error processes required to manually design dispatching rules. A promising direction for GP-HH which is relatively unexplored is to incorporate *ensemble learning* [2] to the GP evolutionary process. Recent research have used coevolutionary GP to evolve ensembles of dispatching rules [3] for DJSS problems, and have found that evolved ensembles outperform standard GP-HH procedures that evolve single dispatching rules. However, they do not seriously consider incorporating a method that aims to diversify the members of the ensembles, which is important part of ensemble learning to improve the generalisation ability of ensembles [2]. In particular, diversity of GP individuals can be improved through niching techniques, such as fitness sharing. By incorporating fitness sharing into the GP process, it may be possible to improve the qualities of the evolved ensembles.

The goal of this paper is to extend an existing coevolutionary GP approach, denoted as Multilevel Genetic Programming for Job Shop Scheduling (MLGP-JSS) [3], that evolves ensembles of dispatching rules for DJSS problems by incorporating a standard fitness sharing algorithm to the GP process. Four different phenotypic distance measures, which calculate the pairwise distances between the individuals in the ensemble, are investigated, and are used by the fitness sharing algorithm. The rules evolved by the niched MLGP-JSS approaches will be compared against the base MLGP-JSS as the benchmark.

2. THE METHOD

The proposed extensions to MLGP-JSS will use discrete-event simulations to both evolve the ensembles of dispatching rules and for evaluation. The simulation procedurally generates the jobs until a certain number of jobs have been processed. The TWT is used as the *performance* measure over both the training and the test sets. To incorporate fitness sharing into the MLGP-JSS process, the following steps are carried out:

1. The fitness sharing is incorporated to MLGP-JSS by adjusting the final fitness of the individual belonging to an ensemble by a penalty factor. In other words, the fitness of an individual is depen-

GECCO'16 Companion July 20-24, 2016, Denver, CO, USA

ACM ISBN 978-1-4503-4323-7/16/07.
DOI: http://dx.doi.org/10.1145/2908961.2908985

Table 1: The performances and the numbers of constituent members for the ensembles evolved by the coevolutionary GP over the test JSS problem instances.

Approach		TWT ($\times 10^5$)	Ensemble Size
$4op$	MLGP-JSS	26.32 ± 2.86	15.1 ± 5.1
	MLGP-D1	28.42 ± 8.78	$6.9 \pm 2.1^{\uparrow}$
	MLGP-D2	27.90 ± 5.19	$9.2 \pm 3.1^{\uparrow}$
	MLGP-D3	27.40 ± 4.41	$10.9 \pm 2.1^{\uparrow}$
	MLGP-D4	26.80 ± 3.91	$10.8 \pm 1.9^{\uparrow}$
$8op$	MLGP-JSS	24.2 ± 4.7	25.32 ± 2.31
	MLGP-D1	27.41 ± 4.89	$9.3 \pm 2.8^{\uparrow}$
	MLGP-D2	$27.24 \pm 4.22^{\downarrow}$	$11.2 \pm 3.2^{\uparrow}$
	MLGP-D3	$27.12 \pm 4.07^{\downarrow}$	$10.3 \pm 2.6^{\uparrow}$
	MLGP-D4	26.80 ± 3.85	$10.8 \pm 2.0^{\uparrow}$

dent on the performance of the individual over the training set and the *diversity* of the individual.

2. The diversity of an individual belonging to an ensemble is calculated using phenotypic distance measures, which are based on the decisions made by the individual as the ensemble is applied to the training instances as a dispatching rule. As an ensemble is applied to a training instance, the individuals in the ensemble carry out *majority voting* at each dispatching decision to select a job to be processed.

3. Four distance measures are investigated for our approach. The distance measures are calculated by comparing the voting processes between the individuals in the ensemble at a dispatching decision. The distance measures are abbreviated to a D# format, e.g., MLGP-D1 denotes niched MLGP-JSS that uses the D1 distance measure. Given that there are L decisions made when generating solution for a problem instance, the distance measures between two individuals ω and ψ in an ensemble E are as follows:

Normalised Priorities (D1): Let $\delta'_\omega(j)$ and $\delta'_\psi(j)$ be the normalised priorities assigned to the selected job by individuals ω and ψ at dispatching decision j. Then $\mathrm{D1}(\omega, \psi) = \sqrt{\frac{1}{L}\sum_{j=1}^{L}(\delta'_\omega(j) - \delta'_\psi(j))^2}$.

Normalised Decision Vector (D2): Let $r_\omega(j)$ and $r_\psi(j)$ be the ranks assigned by ensemble E of the jobs voted by individuals ω and ψ. Then $\mathrm{D2}(\omega, \psi) = \sqrt{\frac{1}{L}\sum_{j=1}^{L}(r_\omega(j) - r_\psi(j))^2}$.

Decision Overlap between Individuals (D3): Let $Avg(\omega, \psi, L)$ be the proportion of overlapping votes between individuals ω and ψ. Then $\mathrm{D3}(\omega, \psi) = 1 - Avg(\omega, \psi, L)$.

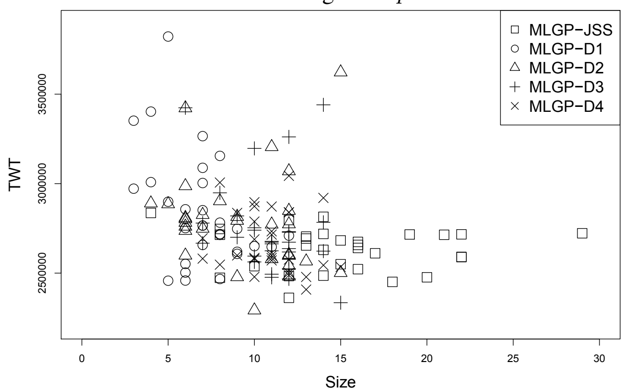
Decision Overlap against Ensemble (D4): Let $Avg(\omega, E, L)$ be the proportion of overlapping votes between individual ω and ensemble E. Then $\mathrm{D4}(\omega, \psi) = 1 - Avg(\omega, E, L)$ for any $\psi \in E$.

3. EXPERIMENTAL DESIGN AND RESULTS

The GP parameters, terminal set and function set are kept consistent as the ones used by Park et al. [3] for both the base and the niched MLGP-JSS approaches. The number of generations is 51. The population size is 1024, 200 groups are breed and 100 groups are retained. The crossover, mutation and reproduction rates are 80%, 10% and 10% respectively. Tournament selection of size 7 is used for the selection procedure in the first generation of MLGP-JSS. In addition, we use an existing dataset from the literature [4] to evaluate the niched GP approaches against the standard GP approaches. There are two training sets ($4op$ and $8op$), each with problem instances that have job arrivals with different numbers of operations per job. The test set contains 20 instances with different configurations for procedurally generating the jobs [4].

The abridged results is in Table 1, where MLGP-JSS is the base MLGP-JSS approach and MLGP-D# is the niched MLGP-JSS approach with distance measure D# (refer to the distance measures in

Figure 1: Figure showing the comparison between the ensemble sizes and the performance of the ensemble over the test set. Ensembles are evolved from training set $4op$.



Section 2). If a niched MLGP-JSS approach performs significantly better than the base MLGP-JSS approach in terms of the performance or the ensemble size, then it is marked with ↑. Otherwise, if it performs significantly worse, then it is marked with ↓. The results show that the niched MLGP-JSS approaches with the different distance measures evolve ensembles with significantly smaller number of constituent members than the base MLGP-JSS approach while specific niched MLGP-JSS approaches have similar performances to the base MLGP-JSS approach. Therefore, further analysis is carried out comparing the size of the ensembles against its performance over the test set. This is shown in Figure 1 for rules evolved over training set $4op$.

From the figure, MLGP-D4 has a moderate correlation coefficient value of -0.48 using Spearman's rank correlation coefficient, whereas the other MLGP-JSS approaches have very weak correlation coefficient values. Therefore, it may be likely that small high quality ensembles can be evolved by coevolutionary GP.

4. CONCLUSIONS AND FUTURE WORK

This paper investigates an approach where a standard fitness sharing algorithm is incorporated into a coevolutionary GP process to evolve ensembles of dispatching rules for DJSS problems. The results show that applying fitness sharing to MLGP-JSS significant reduce the sizes of the ensembles with relatively similar performances over the test set. For future work, additional analysis is required to improve the performance of the ensembles evolved by niched MLGP-JSS, and other coevolutionary GP approaches may need to be investigated.

5. REFERENCES

[1] Branke, J., Nguyen, S., Pickardt, C.W., Zhang, M.: Automated design of production scheduling heuristics: A review. IEEE Transactions on Evolutionary Computation **20**(1) (2016) 110–124

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

[3] Park, J., Nguyen, S., Zhang, M., Johnston, M.: Genetic programming based hyper-heuristics to dynamic job shop scheduling: Cooperative coevolutionary approaches. Proceedings of the 2016 European Conference on Genetic Programming **9594** (2016) 112–127

[4] 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. (2014) 927–934