2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Multi-Objective Island Model Genetic Programming for Predicting the Stokes Flow around a Sphere

Julia Reuter, Pravin Pandey, Sanaz Mostaghim
Otto-von-Guericke-University Magdeburg, Germany
Faculty of Computer Science, Institute for Intelligent Cooperating Systems
{julia.reuter, pravin.pandey, sanaz.mostaghim}@ovgu.de

***Abstract*—This paper is aimed at enhancing the success rate of Genetic Programming (GP) algorithms for symbolic regressions. It is shown that the outcome of GP algorithms over several runs can lead to an optimal solution for such problems, but the success rate, i.e., the number of successful runs, is sometimes small. We address this issue by proposing multi-objective and island model (IM) optimization for GP. We study the influence of various objective functions and IM configurations on the success rates and present 36 algorithm variants, which are tasked with solving two benchmark equations from the fluid mechanics area. This specific benchmark problem has been previously shown to suffer from a low success rate and high variations between the results of multiple runs. Our experiments show a strong influence of the objective functions on the success rate. The additional IM implementation improves the results for some objectives. The algorithm with the highest success rate on the more complex benchmark problem employs both, multiple objectives and IM.**

I. Genetic Programming for Fluid Mechanics

Genetic Programming (GP) for Symbolic Regression (SR) has recently gained considerable importance in identifying equations for applications in engineering and physics. Fluid mechanics problems have proven to be extremely difficult to solve with GP algorithms, due to the complex underlying interactions between particles and fluids. [1]–[3] study several benchmark instances related to the fluid flow around one and two fixed spheres. This flow is governed by the Navier-Stokes equations, which mainly depend upon the so-called Reynolds number Re. Due to the nonlinear nature of the NS equations, there is no universal solution to this problem. However, when Re $\to 0$, these equations can be linearized and solved for the flow around a single spherical particle. Fig. 1 displays this so-called Stokes flow around a single spherical particle.

In this paper, we intend to overcome an issue which appeared frequently when approaching the fluid mechanics benchmark instances introduced in [2] with GP algorithms: decreasing repetition stability and success rate of the algorithm with increasing complexity of the target symbolic models. In other words, when the same algorithm is executed multiple times, it is reliable in identifying comparably simple equations. However, more complex equations showed a large variation in the final result between different runs of the same algorithm. Since high success rates for known equations is an important characteristic of a trustworthy algorithm, we intend to tackle this issue with Island Models (IM). In an IM approach, multiple subpopulations are evolved independently, and several individuals migrated between the subpopulations at specific times [4], [5]. As [2] showed, the choice of objective function also has a significant influence on the success rate of a GP algorithm. We combine both approaches and assess various objective functions and IM configurations to understand how they influence the success rate. The algorithm variants are tasked to solve two fluid mechanics benchmarks from [2], which showed high variation in the results of multiple algorithm executions.

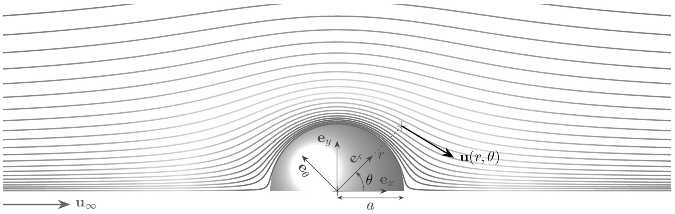


Fig. 1. Streamlines of the Stokes flow around a rigid sphere. u_∞ indicates the direction of the free stream.

$$u_x = u_r \cdot \cos(\theta) - u_\theta \cdot \sin(\theta) \tag{1}$$

$$u_r = u_\infty \cdot \cos(\theta) \cdot \left(1 + \frac{a^3}{2 \cdot r^3} - \frac{3 \cdot a}{2 \cdot r}\right) \tag{2}$$

Eq. 1 converts the fluid velocity from spherical coordinates (u_r, u_θ) to Cartesian coordinates (u_x), as a function of the polar angle θ. Eq. 2 describes the velocity field around a spherical particle with radius a, subject to the far-field velocity u_∞ in a spherical coordinate system (r, θ), whose origin is located at the center of the particle. This paper extends the work of [2] by combining multi-objective GP with IM GP.

II. Background and Related Work

A. Genetic Programming

GP was first introduced by Koza and stems from the family of evolutionary algorithms (EA) [6]. It is a suitable method for SR, i.e., identifying equations from data. An equation is usually represented as a parse tree. The inner nodes consist of functions from the function set $\mathcal{F}$, where each function has a pre-defined number of input arguments. Features and constants build the terminal set $\mathcal{T}$, which are located at the leaf nodes.

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project 466092867 – within the Priority Programme "SPP 2331: Machine Learning in Chemical Engineering".

978-0-7381-4408-5/23/$31.00 ©2023 IEEE

Initially, a random population of trees is created and iteratively refined through evolution. Crossover swaps branches between parent trees to form new children. Mutation is applied to induce minor changes in an individual, usually on the node-level, such as randomly replacing a function with another.

The main goal is to find the mathematical model that best fits the given dataset. The fitness of a candidate solution is determined by comparing the prediction to the actual data points, utilizing problem-dependent objective functions. Good solutions according to the defined objectives are more likely to be selected as parents to form new children, and to survive to the next generation. The evolutionary process is repeated until a stopping criterion is reached.

B. Multi-objective Optimization

The main objective in GP is to identify an equation that perfectly predicts the target variable from the given input data. To this end, an error function between the predicted output and the target variable is employed as an objective, such as the mean squared error (MSE) or the rooted mean squared error (RMSE). The latter is often preferred as it has the same unit as the target variable, making it easier to classify the magnitude of the error.

EAs, and thus also GP, are predestined to optimize multiple criteria at the same time [7]. When dealing with multi-objective optimization, one can represent the problem as:

$$\begin{aligned} \min \quad & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_m(\mathbf{x}))^T \\ s.t. \quad & \mathbf{x} \in \Omega \end{aligned} \tag{3}$$

The search space Ω maps to the m-dimensional objective space $\mathcal{M}$. In GP, Ω spans the space of all possible models that can be created from $\mathcal{T}$ and $\mathcal{F}$. The objective functions $f_j(\mathbf{x}), j \in \{1, \dots, m\}$ are in conflict with each other, so that different solutions satisfy the objectives to varying degrees. Thus, more than one optimal solution to such problems exists, a so-called set of Pareto-optimal solutions. The concept of Pareto-dominance is used to obtain such solutions. A solution $\mathbf{x}_1$ is said to dominate the solution $\mathbf{x}_2$, if the following conditions are met [7]: (1) The solution $\mathbf{x}_1$ is no worse than $\mathbf{x}_2$ in all objectives, i.e. $f_j(\mathbf{x}_1) \leq f_j(\mathbf{x}_2)$ for all $j = 1 \dots m$. (2) The solution $\mathbf{x}_1$ is strictly better than $\mathbf{x}_2$ in at least one objective, i.e. $f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2)$ for at least one $j = 1 \dots m$. Pareto-optimal solutions are not dominated by any other solution.

Recent work addressing the same benchmark functions suggests, that GP algorithms can benefit from additional objectives to achieve better solutions in the final population [2]. They employed the maximum absolute error (MAE) as the first objective to be minimized, and assessed combinations with additional objectives. Specifically, a size, a dimension-penalty objective and a correlation objective were proposed. The size of an equation is determined by the number of nodes of a tree. More concise solutions tend to generalize better and are less prone to overfitting the data. For SR problems from the engineering and physics area, it is particularly important to identify equations that are conformal with physical laws. To this end, different approaches have been presented, such as strongly-typed GP [8], grammar-based GP [9], [10], as well as multi-objective unit-aware GP [11]. The latter uses a dimension penalty as a minimization objective, where each unit-violating operation within an individual is penalized.

C. Island Models

The goal of using IMs for EAs is to explore different areas of the search space through evolution on independent islands, and increase diversity within the subpopulations through migration in order to prevent the algorithm from premature convergence. Furthermore, the use of subpopulations allows for a parallel implementation of the algorithm, resulting in significant speed-ups compared to the single population algorithm [4]. To this end, multiple subpopulations are evolved in parallel, and migrate several individuals at specific times during the evolution. The main hyperparameters of such algorithms are the number of islands m, the number of individuals to migrate k, and a communication strategy between the islands. Furthermore, the migration rate or number of migrations over the total number of generations plays an important role. Generally, the migration topologies are not GP-specific but can be inspired from the EA area. Different topologies have been proposed and explored [5], [12], [13], most prominently ring, grid, mesh and random topology. While ring, grid, and mesh topology specify a specific interaction pattern between the islands, the random approach migrates individuals from a randomly chosen island to another. A modification of the random topology has recently been presented in [14], which uses an archive of best solutions among all combined subpopulations. Random individuals from a subpopulation are exchanged by randomly selected solutions from the archive.

Making use of the building block hypothesis in GP, Ono et al. propose an adaptive migration strategy based on frequent subtrees, which outperforms the original algorithm [15], [16]. The SR benchmark equation in that paper was approached in a single-objective manner using an error objective. While some papers address the dynamics of the evolution induced by IMs [13], [17], our focus is solely output-oriented, i.e., increasing the success rate of a GP algorithm to achieve a perfect solution. To the best of our knowledge, Fillon et al. have first addressed the issue of maximizing the success rate with IMs [18].

III. Proposed Methods

In this paper, we assess the impact of combinations of different objective functions and IM configurations on enhancing the success probability of an algorithm. In the following, we introduce algorithmic components that will be examined in our experiments. From these experiments on known benchmark equations, our goal is to deduce optimal algorithmic configurations. These can be utilized in the future to discover unknown equations.

A. Objective Functions

Inspired by the approach of Zille et al. [2], we adopt a multi-objective perspective in our methodology. Our primary objective, denoted as f_1, minimizes the error between the prediction and the target variable. To quantify this error, we employ the RMSE, a widely used fitness measure for GP, which emphasizes larger errors with greater penalties. We want to point out that the error objective used in [2] is, contrary to ours, the MAE. Their preliminary experiments demonstrated that MAE yielded superior performance on the given benchmark equations compared to RMSE. However, the MAE may be used with caution when unknown equations are discovered from data. Experimental data often contains noise and outliers, and MAE tends to put higher emphasis on minimizing the error on the outliers, rather than finding an equation that performs generally well across the entire dataset. Since we aim at deducing a robust algorithm configuration to discover unknown equations in the future, we employ the RMSE in our experiments.

We adopt the correlation objective as well as the dimension penalty from [2], thus we will only shortly introduce them here. We incorporate the rank-based Spearman correlation as our second objective f_2 in the evolutionary process. The purpose is to enable individuals with a strong correlation with the target variable to survive to the next generation, even when they perform poorly on the error objective f_1. This approach enables promising individuals to undergo refinement, and guiding the search towards more accurate solutions. The formulation $f_2 = 1 - |\rho|$ is used as a minimization objective, where ρ represents the Spearman correlation coefficient, ranging from -1 to 1.

The third objective is a dimension penalty which penalizes individuals that execute non-physical operations. When a unit-violating operation occurs, the penalty value n_{viol} is increased by 1.0 for all operations within an individual. Moreover, this objective has the purpose to align the output unit of a solution with the target unit, which is in our case `meters/second` = $\mathtt{m}^1 \cdot \mathtt{s}^{-1}$. The Manhattan distance between the exponents of the SI-units of the predicted and target units is added to the penalty value, resulting in $f_3 = n_{\text{viol}} + ||\hat{u}_{SI} - u_{SI}||_1$. Both, f_2 and f_3, are supporting objectives, i.e., we assess whether the inclusion of these objectives improves the performance with regard to f_1.

B. Island Model Configurations

As pointed out in Sec. II-C, IM algorithms have numerous hyperparameters which potentially affect their performance. An extensive study in a grid-search-fashion over multiple values for all hyperparameters exceeds the scope of this paper. Therefore, we will adopt some hyperparameters from related research. Motivated by [14], we investigate the performance of archive-based migration compared to non-archive based migration. To this end, we propose two migration topologies:

- Archive-based migration (A): The best k individuals of all subpopulations build an archive of overall best solutions. For each subpopulation of size m, we replace k individuals at random with k individuals from the archive. Fig. 2 displays the archive-based migration topology.

Fig. 2. Archive-based migration topology

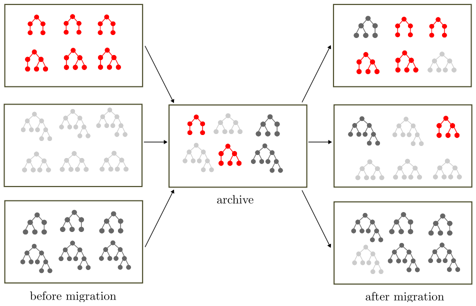
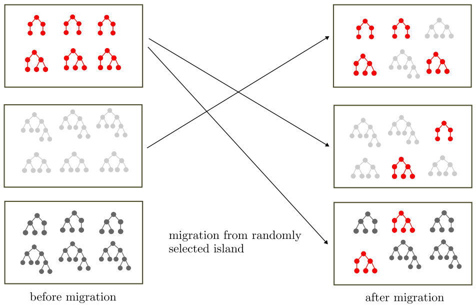


Fig. 3. Random migration topology



- Random migration (R): In the random migration topology, the subpopulation from which individuals migrate to another is chosen at random. The k worst individuals of the respective island are replaced with the k best individual of the randomly selected island. Due to the random nature of the migration, multiple subpopulations can receive individuals from the same island (Fig. 3).

Each migration can increase the diversity of solutions within a subpopulation. Usually, a fixed migration rate is used in an IM algorithm, so that migrations are uniformly distributed over the algorithm runtime. In preliminary trials, we observed that some algorithms perform better when migrations are only performed in the first two thirds of the algorithm runtime. The last third without migration might facilitate the algorithm to converge without further disturbance introduced by migration. Therefore, in addition to the migration topology, we also investigate on the distribution of migrations over the evolution.

Alg. 1 outlines the proposed IM algorithm. The migration schedule m_{sched} contains the generation indices in which migrations are performed. The migration function in line 12 calls either the archive-based or random migration topology. The algorithm returns a hall of fame (HOF), which contains the set of all non-dominated solutions.

IV. Experiment Setup

A. Benchmark Datasets

We conduct experiments to assess the effect of various combinations of objective functions and IM configurations on the success rate of a GP algorithm. To this end, we employ the fluid mechanics benchmark functions defined in Eq. 1 and 2. We choose these equations because they have previously

Algorithm 1 Proposed Island Model Algorithm

```
Input: training data $X$, number of subpopulations $m$, subpopulation size $p$, number of generations $n$, migration schedule $m_{\text{sched}}$
Output: hall of fame $H$
$pops \leftarrow$ set of $m$ random initial subpopulations, each of size $p$
evaluate($pops$)
$H \leftarrow$ updateHOF($pops$)
for $gen = 1 \dots n$ do
    for $i = 1 \dots m$ do
        $parents \leftarrow$ select($pops[i]$)
        $offspring \leftarrow$ reproduce($parents$)
        evaluate($offspring$, $X$)
        $pops[i] \leftarrow$ updatePopulation($\cup(pops[i], offspring)$)
    end for
    if $gen \in m_{sched}$ then
        $pops \leftarrow$ migrate($pops$)
    end if
    $H \leftarrow$ updateHOF($pops$)
end for
return $H$
```

TABLE I
BENCHMARK EQUATIONS

Eq.	Training Features	Target Variable	# Samples
1	u_r, u_θ, θ	u_x	366
2	u_∞, a, r, θ	u_r	366

shown high variations in the results between multiple runs [2]. Tab. I summarizes the benchmark datasets. The dataset is split in a ratio of 0.8 for training and 0.2 for testing.

B. Algorithm Variants

We compare the performance of one single-objective and three multi-objective functions as defined in Sec. III-A with each other: f_1, $f_1 f_2$, $f_1 f_3$, $f_1 f_2 f_3$. Furthermore, we assess the impact of different IM configurations. As mentioned earlier, we cannot capture the entire parameter space of IM algorithms in this paper. We compare the performance of a single population algorithm, denoted by S, to IM algorithms with $m \in [5, 10]$ subpopulations. Similar configurations have been used in related works [16], [17]. The migrations are executed according to the random (R) and archive-based (A) migration topologies as defined in Sec. III-B. The number of migrations is set to 10, which are distributed over the algorithm runtime according to two schemes. In the first variant, they are distributed uniformly, denoted by D0. In the second variant, they only happen in the first two thirds of the algorithm runtime, denoted by D1. To facilitate comparison between the single population and IM algorithms, the population size of 2000 individuals is evenly distributed among the respective number of islands. A subpopulation thus constitutes $\frac{2000}{m}$ individuals. The proportion of migrating individuals is set to 0.035, which was also used in [14].

TABLE II
GP ALGORITHM PARAMETERS

Parameter	Settings
Number of Generations	800
Reproduction Scheme	$\mu + \lambda$
Selection Mechanism	NSGA-II selection
Initialization Method	Half Full, Half Grow
Crossover Probability p_c	0.5
Mutation Probability p_m	0.5
Crossover	One-point, Leaf-biased ($p_{\text{leaf}} = 0.9$) (chosen at random)
Mutation	Uniform, Insert, Shrink, Node Replacement (chosen at random)
Max. Tree Length	30

In summary, we test four objective functions with one single population and eight IM configurations, which makes a total of 36 algorithm variants. All experiments employ the function set $\mathcal{F} = \{+, -, \cdot, \circ^2, \circ^3, \frac{1}{\circ}, \sin(\circ), \cos(\circ), \sqrt{\circ}\}$ and the terminal set $\mathcal{T} = \mathcal{C} \cup \{\text{training features}\}$, where $\mathcal{C} = \{4, 3, 2, 1, \frac{1}{2}, \frac{1}{4}\}$. The experiments are run for 800 generations. To compare the effects of the assessed algorithmic variants, we use the same random seed in each of the 31 algorithm executions across all algorithm configurations. Tab. II summarizes additional settings of the GP algorithms used in our experiments, which are implemented in the Python library `deap` [19].

V. RESULTS AND ANALYSIS

In the following, we provide a comprehensive analysis of the experimental results. Tab. III gives an overview of the number of successfully solved runs out of 31 runs. To this end, we first determine the best solution per run to be the individual in the HOF with the lowest RMSE on the test dataset. A run is counted as solved, when the best individual of a run has an error objective $f_1 < 1e-05$ on the test dataset. For Eq. 1, the single-objective single population algorithm solved five runs successfully. The single-objective IM algorithm solved slightly more runs successfully, when the individuals were distributed over ten islands. A similar behavior can be observed for the objectives $f_1 f_2$. To the contrary, objective $f_1 f_3$ achieves higher success rate when only five islands are employed. The objective with the highest success rate is $f_1 f_2 f_3$, where no clear difference between five or ten islands can be observed. The overall most successful configuration solved 25 runs.

Eq. 2 is more complex and involves a higher number of computations. This is also reflected in the success rate of the single-objective single population algorithm, which did not solve a single run successfully. Objective $f_1 f_2$ achieves slightly higher success rates, and $f_1 f_2 f_3$ solved seven runs with the single population algorithm and 13 with an IM algorithm. $f_1 f_2 f_3$ tends to perform slightly better, when the population is distributed over ten islands. Objective $f_1 f_3$ did not improve the success rate compared to the single-objective algorithm. Generally, at least one IM configuration solved more runs successfully compared to the single population approach of the same objectives. However, no clear best performing IM configuration can be identified.

TABLE III

COUNTS OF SUCCESSFUL RUNS FOR THE TWO BENCHMARK INSTANCES AND COMBINATIONS OF OBJECTIVES AND IM CONFIGURATIONS. A RUN IS SUCCESSFUL, WHEN THE ERROR f_1 OF THE BEST PERFORMING INDIVIDUAL OF THE FINAL POPULATION IS BELOW $1e-05$ ON THE TEST DATASET.

Eq.	# Isl.	Top.	Distr.	f_1	f_1f_2	f_1f_3	$f_1f_2f_3$
1	1	S	-	5	11	11	15
	5	A	D0	4	17	12	22
		A	D1	4	14	12	19
		R	D0	4	16	12	25
		R	D1	4	14	13	20
	10	A	D0	7	21	5	21
		A	D1	6	21	5	21
		R	D0	6	21	7	23
		R	D1	6	20	7	23
2	1	S	-	0	1	0	7
	5	A	D0	0	1	0	11
		A	D1	0	1	0	8
		R	D0	0	3	0	10
		R	D1	0	1	1	7
	10	A	D0	0	2	0	13
		A	D1	0	1	0	13
		R	D0	0	2	0	10
		R	D1	0	2	0	10

To validate our observations, we conduct statistical tests on the error distributions of the algorithm variants. For each row, i.e., algorithm variant, the best performing objective in terms of RMSE is the baseline method and marked in bold. For each column, i.e., objective, and benchmark instance, the single population algorithm as a baseline is compared to all IM variants. The asterisk symbol indicates that an IM configuration performs significantly better than the respective single population algorithm on this benchmark instance. We use the one-sided Mann-Whitney-U statistical test for each algorithm to test whether its performance is significantly worse than the baseline method. The level of significance is $\alpha = 0.05$. The results are displayed in Tab. IV.

On Eq. 1, the best performing objectives for the single population algorithm are f_1f_3 and $f_1f_2f_3$. The IM variants perform best when f_1f_2 or $f_1f_2f_3$ are used, which outperform the other two objectives but are not significantly different from each other. Interestingly, most IM variants outperform the single population algorithm when f_1f_2 is used. However, only one IM algorithm shows better performance when the dimension penalty is added as an objective, i.e., when $f_1f_2f_3$ is employed. Taking a look at the results of Eq. 2, the objective $f_1f_2f_3$ outperforms the other objectives for all IM settings, except for the single population algorithm. Furthermore, almost all IM configurations outperform the single population algorithm when $f_1f_2f_3$ is used as an objective. For the other three objective variants, most IM algorithms did not show a significantly better result than the single population algorithm.

In addition to the results in Tab. IV, we perform a Kruskal-Wallis test on the IM variants for each objective and benchmark instance. No significant differences between the IM variants could be identified for all objectives and benchmark instances. The effects of the migration topology and the distribution of migrations over the algorithm runtime are negligible.

Altogether, we can observe for the less complex Eq. 1, that the IM algorithm only showed significantly better results when the objective f_1f_2 was utilized. The more difficult to solve Eq. 2 benefits from both, the use of a multi-objective algorithm $f_1f_2f_3$ and the distribution of the population on multiple islands. In cases where the results improved by the implementation of an IM algorithm, this happened independent of the exact choice of the IM hyperparameters.

It is of great interest to which extent the proposed methods can help to discover new equations from data in the future. We assume that new equations are likely to involve a high number of operations and therefore closer to the difficulty of Eq. 2. First, preferring RMSE over MAE as the first objective can help to discover good equations even on noisy data. Based on the results of this paper, we strongly recommend using correlation as supporting objective f_2. The satisfaction of physical laws is required for most engineering applications, which is why we suggest including f_3. We want to point out that our benchmark equations did not contain unknown coefficients, which will likely be the case when discovering new equations. Since coefficient units are not known beforehand, an adapted dimension penalty is required, which takes this into account. Altogether, we recommend using the objectives $f_1f_2f_3$ as well as an IM configuration for such GP algorithms, especially since the IM comes at no additional computational cost compared to the single population algorithm.

VI. CONCLUSION AND FUTURE WORK

This paper aimed at enhancing the success rate of genetic programming algorithms for symbolic regression. We presented a comparison of different objective functions and configurations of island model algorithms. The objectives comprise an error function, a correlation coefficient and a dimension penalty. We tested the performance of these objec-

TABLE IV
RMSE VALUES (MEAN ± STANDARD DEVIATION) OVER 31 RUNS FOR THE TWO BENCHMARK INSTANCES AND ALL COMBINATIONS OF IM CONFIGURATIONS AND OBJECTIVES. NUMBERS IN BOLD INDICATE THE BEST RESULT FOR EACH ROW, I.E., WHICH OBJECTIVE PERFORMED BEST FOR THE SPECIFIC IM CONFIGURATION. THE ASTERISK SYMBOL INDICATES WHETHER AN IM CONFIGURATION OUTPERFORMS THE RESPECTIVE SINGLE POPULATION VARIANT OF THE SAME OBJECTIVES AND BENCHMARK INSTANCE WITH STATISTICAL SIGNIFICANCE.

Eq.	# Isl.	Top.	Distr.	f_1	f_1f_2	f_1f_3	$f_1f_2f_3$
1	1	S	-	6.191e-03 ± 5.682e-03	2.777e-03 ± 2.510e-03	**2.559e-03** ± 3.113e-03	**1.152e-03**± 1.593e-03
	5	A	D0	5.921e-03 ± 4.178e-03	**1.378e-03** ± 1.890e-03 ∗	3.781e-03 ± 3.894e-03	**2.822e-04** ± 5.693e-04
		A	D1	5.042e-03 ± 3.483e-03	**1.838e-03** ± 2.214e-03	3.457e-03 ± 3.401e-03	**6.255e-04** ± 1.761e-03
		R	D0	5.723e-03 ± 4.495e-03	**1.488e-03** ± 1.921e-03 ∗	3.732e-03 ± 4.215e-03	**3.059e-04** ± 8.645e-04 ∗
		R	D1	5.443e-03 ± 3.924e-03	**1.373e-03** ± 1.763e-03 ∗	3.274e-03 ± 3.435e-03	**4.157e-04** ± 7.387e-04
	10	A	D0	3.900e-03 ± 4.058e-03	**7.297e-04** ± 1.282e-03 ∗	4.990e-03 ± 3.273e-03	**3.487e-04** ± 8.653e-04
		A	D1	4.188e-03 ± 3.931e-03	**7.466e-04** ± 1.596e-03 ∗	5.378e-03 ± 3.500e-03	**3.317e-04** ± 8.716e-04
		R	D0	5.233e-03 ± 4.577e-03	**6.842e-04** ± 1.233e-03 ∗	3.902e-03 ± 3.332e-03	**4.110e-04** ± 1.114e-03
		R	D1	5.159e-03 ± 4.447e-03	**7.869e-04** ± 1.396e-03 ∗	4.568e-03 ± 3.615e-03	**4.647e-04** ± 1.126e-03
2	1	S	-	3.830e-03 ± 2.390e-03	**8.796e-04** ± 8.122e-04	4.201e-03 ± 2.637e-03	**7.766e-04** ± 8.280e-04
	5	A	D0	3.366e-03 ± 1.469e-03	6.993e-04 ± 6.384e-04	3.535e-03 ± 1.816e-03	**2.970e-04** ± 3.448e-04 ∗
		A	D1	3.006e-03 ± 1.725e-03	7.466e-04 ± 6.226e-04	3.270e-03 ± 2.160e-03	**2.494e-04** ± 2.655e-04 ∗
		R	D0	3.544e-03 ± 1.847e-03	9.281e-04 ± 7.806e-04	2.735e-03 ± 1.658e-03 ∗	**3.510e-04** ± 5.497e-04 ∗
		R	D1	3.415e-03 ± 1.671e-03	1.170e-03 ± 9.011e-04	2.785e-03 ± 1.527e-03 ∗	**4.153e-04** ± 4.655e-04
	10	A	D0	3.468e-03 ± 2.015e-03	6.298e-04 ± 5.999e-04	3.417e-03 ± 1.608e-03	**2.658e-04** ± 6.519e-04 ∗
		A	D1	3.288e-03 ± 1.980e-03	7.637e-04 ± 7.666e-04	3.343e-03 ± 1.706e-03	**2.691e-04** ± 3.139e-04 ∗
		R	D0	3.146e-03 ± 1.514e-03	6.788e-04 ± 5.419e-04	3.224e-03 ± 1.592e-03	**3.540e-04** ± 3.907e-04 ∗
		R	D1	3.113e-03 ± 1.671e-03	7.892e-04 ± 7.575e-04	3.165e-03 ± 1.536e-03	**2.796e-04** ± 3.234e-04 ∗

tives on a varying number of islands, two migration topologies and two distributions of the migrations over the algorithm runtime. The 36 algorithm variants were tasked with solving two benchmark equations from the fluid mechanics area, which previously showed high variations between the results of multiple runs. The results of our experiments show a strong influence of the objective function on the success rate of the algorithm. For some objectives, the results improved further when in island model approach was used compared to a single population algorithm. These objectives were also the best performing objectives overall. No significant differences were found between the IM configurations, suggesting that results for some targets improve as long as migration is performed.

Our results build a promising starting point for future research directions. On the one hand, our proposed algorithms can be applied to a more diverse set of benchmark equations to further verify our recommendation for discovery of new equations. In addition, approaches to account for unknown units of the coefficients during dimensional analysis form an important extension of this paper.

REFERENCES

[1] H. Zille, S. Mostaghim, F. Evrard, and B. van Wachem, "Unit-aware multi-objective genetic programming for the prediction of the stokes flow around a sphere," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM, Jul. 2021, pp. 327–328.

[2] H. Zille, F. Evrard, J. Reuter, S. Mostaghim, and B. Wachem, "Assessment of multi-objective and coevolutionary genetic programming for predicting the stokes flow around a sphere," in *EUROGEN '21 Proceedings*, 2021, p. 190.

[3] J. Reuter, M. Cendrollu, F. Evrard, S. Mostaghim, and B. van Wachem, "Towards Improving Simulations of Flows around Spherical Particles Using Genetic Programming," in *CEC '22 Proceedings*, 2022, pp. 1–8.

[4] D. Andre and J. R. Koza, "A parallel implementation of genetic programming that achieves super-linear performance," *Information Sciences*, vol. 106, no. 3-4, pp. 201–218, May 1998.

[5] W. F. Punch, "How effective are multiple populations in genetic programming," in *Genetic Programming 1998: Proceedings of the Third Annual Conference*. Morgan Kaufmann, 22, pp. 308–313.

[6] J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection*. MIT press, 1992, vol. 1.

[7] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. USA: Wiley, 2001.

[8] S. Wappler and J. Wegener, "Evolutionary unit testing of object-oriented software using strongly-typed genetic programming," in *GECCO '08 Proceedings*, 2006, p. 1925–1932.

[9] R. I. Mckay, N. X. Hoai, P. A. Whigham, Y. Shan, and M. O'neill, "Grammar-based genetic programming: a survey," *Genetic Programming and Evolvable Machines*, vol. 11, no. 3-4, pp. 365–396, 2010.

[10] A. Ratle and M. Sebag, "Grammar-guided genetic programming and dimensional consistency: application to non-parametric identification in mechanics," *Applied Soft Computing*, vol. 1, no. 1, pp. 105 – 118, 2001.

[11] D. Li and J. Zhong, "Dimensionally aware multi-objective genetic programming for automatic crowd behavior modeling," *ACM Transactions on Modeling and Computer Simulation*, vol. 30, no. 3, pp. 1–24, 2020.

[12] F. Fernández, M. Tomassini, W. F. Punch, and J. M. Sánchez, "Experimental Study of Multipopulation Parallel Genetic Programming," in *Genetic Programming*, 2000, vol. 1802, pp. 283–293.

[13] F. Fernández, M. Tomassini, and L. Vanneschi, "Studying the Influence of Communication Topology and Migration on Distributed Genetic Programming," in *Genetic Programming*, 2001, vol. 2038, pp. 51–63.

[14] M. Cranmer, "Interpretable machine learning for science with pysr and symbolicregression.jl," *ArXiv*, vol. abs/2305.01582, 2023.

[15] K. Ono, Y. Hanada, M. Kumano, and M. Kimura, "Island model genetic programming based on frequent trees," in *CEC '13 Proceedings*, 2013, pp. 2988–2995.

[16] ——, "Enhancing Island Model Genetic Programming by Controlling Frequent Trees," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, no. 1, pp. 51–65, 2019.

[17] F. F. Fernández, M. Tomassini, and L. Vanneschi, "An Empirical Study of Multipopulation Genetic Programming," *Genetic Programming and Evolvable Machines*, vol. 4, pp. 21–51, 2003.

[18] C. Fillon and A. Bartoli, "A Divide & Conquer Strategy for Improving Efficiency and Probability of Success in Genetic Programming," in *Genetic Programming*, 2006, vol. 3905, pp. 13–23.

[19] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary algorithms made easy," *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, 2012.