

# An Investigation on Multi-Objective Fish Breeding Program Design

Fergus Currie, Yi Mei, Mengjie Zhang  
School of Engineering and Computer Science  
Victoria University of Wellington  
Wellington, New Zealand  
{currieferg,yi.mei,mengjie.zhang}@ecs.vuw.ac.nz

Maren Wellenreuther, Linley Jesson  
Plant and Food Research  
Nelson, New Zealand  
{maren.wellenreuther,linley.jesson}@plantandfood.co.nz

**Abstract**—This paper studies the multi-objective fish breeding program design (MO-FBPD), which is an important problem with significant impact in the aquaculture industries. For the first time, we formulated the problem into a multi-objective optimisation problem, including the design of the decision variables, objective functions and constraints. We also developed a simulator for the fish breeding process to facilitate research and analysis. Then, we applied the well-known NSGA-II to solve the MO-FBPD problem, analysed the performance of NSGA-II, and the distributions of the solutions obtained by the algorithm. The results shown are promising. First, it is observed that the breeding program, represented as the top percentage of the fish selected for mating, can generalise to different fish population size. In other words, the trained breeding programs can be applied to future breeding processes with any fish population size. Second, it is observed that the long-term and short-term gains of the fish population are conflicting with each other, and they follow a linear relationship. Finally, if the selection strength is too greedy, it may negatively affect even the short-term gain. This work is a promising preliminary study in this problem domain, which can pave the way for further research in MO-FBPD.

## I. INTRODUCTION

Breeding program design is an important problem in agriculture and aquaculture industries. A breeding program controls the breeding of a population with the aim of a breeding new individuals with improved production traits such as fast growth and size.

In this paper, we focus on studying the fish breeding program design. A fish breeding program can be represented as a set of decisions. Selection strength is the proportion of fish culled between generations, and is the decision variable used in this research. Strong selection for well performing individuals increases mean productive traits in the population but reduces genetic diversity.

A well designed program allows desired productive gains to be made while maintaining genetic diversity. This allows for population size to be easily minimised, reducing cost without damaging either productive gains or diversity. This problem has inherent biological trade-off between productive traits and genetic diversity, so the problem can be formulated as multi-objective problem, allowing for an analytic balance between objectives rather than manually designed heuristic approaches.

Multi-Objective Optimisation (MOO) is a well researched field, which has had extensive application to optimisation prob-

lems. In recent years evolutionary multi-objective optimisation (EMO) [1] has become the state of the art technique for solving this task, and has been applied to a wide range of problems including vehicle routing [2], fault diagnosis [3], water distribution [4] and a multitude of design problems [5] [6] [7]. However, there is no existing research on multi-objective optimisation applied to animal breeding, although grains have had some attention [8], [9]. This specific area of research is lacking in data and requires a simulation to be developed.

This paper aims to investigate the MO-FBPD problem, with the following specific research objectives.

- Develop the simulation for the fish breeding process based on the SLiM [10] software. The fish gene representation, breeding process and simulation configurations (e.g., population size, mating rate, selection strength, breeding duration) are designed.
- Formulate the fish breeding program design as a multi-objective optimisation problem, including the definition of the decision variables and objective functions (i.e., *fish quality* and *genetic diversity*).
- Apply NSGA-II to solve the multi-objective fish breeding program design problem and investigate the performance of the obtained breeding programs on the training simulations and the generalisation to the unseen test simulations.

The rest of the paper is organised as follows. Section II gives the background on the breeding program design problems and the multi-objective optimisation. Then, Section III describes the MO-FBPD problem defined in this study. Section IV shows how to apply NSGA-II to solve the proposed MO-FBPD problem. Section V is the experimental studies, and results and analysis. Finally, Section VI concludes the paper and shows some potential future directions.

## II. BACKGROUND

### A. Breeding Program Design

Besson et al. [11] investigated the relationship between feed conversion ratio and growth rate and how the relationship should be applied to fish breeding program design. Feed conversion ratio (FCR) is a standard breeding metric of efficiency of livestock. It is calculated as the mass of feed vs

output meat. Three different breeding methods are evaluated; maximising growth, maximising growth and economic return, and maximising growth and minimising eutrophication. Both two-trait breeding methods outperform the baseline in terms of economic return and eutrophication respectively. However no trade-off between these traits is investigated.

Holtmark et al. [12] studied how genetic relationships within initial breeding populations affect diversity and trait gain. Simulation is used to gather data, trialling different configurations of sub-populations for each breeding program. Other parameters of the breeding program are kept constant; selection is carried out for 10 generations, maximising genetic trait gain, at a fixed rate of inbreeding. Inbreeding rate uses the theory of optimal contribution selection [13]. A breeding program with sub-populations was found to positively impact genetic diversity. Within the simulation an increase from one sub-population to four increased genetic diversity by 7%.

A simulation can be used as a surrogate to a true fish breeding program. SLiM [10] is a genetic simulation framework, built upon Edios Scribe. SLiM allows control over genomic and evolutionary components of a population. Genomic components include chromosome length, mutation types and mutation rates. Evolutionary components include selection, population size and number of generations. SLiM is effective for modeling evolutionary problems. This paper [14] on fish mating design reviewed the relationship between genetic variance and production traits. A systematic approach for finding an optimal breeding program is not produced. These experiments are performed in a simulation. Competing objectives are mentioned but multi-objective is not considered. Multi-objective breeding strategies for sustainable food improvement [9] studied the use of multi-objective approaches for wheat breeding. It optimised the trait gain with genetic variation. It presented several approaches and compared them to standard breeding schemes through data and simulations. At the end of long term simulations trait gains were 20%-30% greater for multi-objective techniques compared with traditional approaches.

### B. Evolutionary Multi-Objective Optimisation

A Multi-objective Optimisation Problem (MOP) [15]–[17] can be modelled as follows.

$$\min F(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x})), \quad (1)$$

$$s.t. : \mathbf{x} \in \Omega, \quad (2)$$

where  $\mathbf{x}$  is the decision vector, and  $\Omega$  is the decision space (the set of values that each variable can take). There is no single global optimal  $\mathbf{x}$  that achieves the optimal value of all the objectives. Note that we can easily convert any maximisation into minimisation by negating the objective function.

**Definition 1** (Dominance relation). Given two solutions  $\mathbf{x}_1$  and  $\mathbf{x}_2$ ,  $\mathbf{x}_1$  *dominates*  $\mathbf{x}_2$  (or  $\mathbf{x}_2$  *is dominated by*  $\mathbf{x}_1$ ) if (1)  $\forall k \in \{1, \dots, m\}$ ,  $f_k(\mathbf{x}_1) \leq f_k(\mathbf{x}_2)$  and (2)  $\exists k \in \{1, \dots, m\}$ ,  $f_k(\mathbf{x}_1) < f_k(\mathbf{x}_2)$ .

**Definition 2** (Pareto optimality). A solution  $\mathbf{x}^*$  is a *Pareto optimal solution* if there is no other solution in  $\Omega$  that dominates  $\mathbf{x}^*$ .

The *Pareto optimal set* is the set of all the *Pareto optimal solutions* in  $\Omega$ .

The *Pareto front* is the projection (objective vectors) of the Pareto optimal set in the objective space.

The aim of a MOP is to find the Pareto optimal set and Pareto front rather than a single global optimum. This makes it much more challenging than single-objective optimisation as it needs to consider various factors such as coverage and diversity that single-objective optimisation does not consider. Evolutionary Algorithms (EAs) are promising techniques to solve MOP due to its population-based search mechanism, which makes it natural to optimise a set of solutions simultaneously.

There are three main design issues in Evolutionary Multi-objective Optimisation (EMO) [18], which are (1) fitness assignment; (2) diversity preservation and (3) elitism. For fitness assignment, one needs to assign a scalar fitness value to the individuals for the parent and environmental selection. For diversity preservation, unlike in single-objective optimisation where all the individuals tend to converge to the global optimum, it is important in MOP to make the individuals in the population diversely distributed so they can have a good spread and cover the Pareto front well. The elitism means that the non-dominated solutions should not be lost in the later stage of the search process.

So far, the existing EMO algorithms can be categorised into three main groups. The first group is the indicator-based approaches, which use the performance measure indicators such as hypervolume (HV) to assign fitness to the individuals. Examples include [19]–[21].

The second group is based on the Pareto dominance relation, and selects individuals based on the Pareto dominance relation. A famous algorithm belonging to this category is the NSGA-II [22]. Specifically, at each generation of NSGA-II, an offspring population is first generated from the current population. Then a new population is formed by merging the current and offspring populations together. After that, a non-dominated sorting is conducted (with complexity of  $O(MN^2)$ ), and all the individuals in the merged population are divided into several “fronts” based on their dominance relations with others. That is, all the non-dominated individuals in the merged population are placed into the first front. Then, all the remaining individuals that are dominated by only the individuals in the first front are placed into the second front. This way, the individuals in each front are dominated by only those in the previous fronts. In addition, to distinguish the individuals in the same front, the *crowding distance* metric is calculated so that the individuals in less crowding areas (larger crowding distances) are preferred. Other well-known algorithms include SPEA2 [23].

The third group contains the decomposition-based algorithms. These algorithms decompose the original MOP into

a set of single-objective optimisation sub-problems (each corresponding to a region/point of the Pareto front). To breed offspring for each sub-problem, the parents are selected from a subset of the population containing the individuals with high fitness for the corresponding sub-problem. A representative algorithm in this category is MOEA/D [24]

EMO algorithms are criticised for their large number of function evaluations [25]. For problems with expensive function evaluations this can make traditional algorithms infeasible, expensive or slow. Many techniques have been suggested to address this [25]–[27]. They can be grouped into three main categories; problem approximation, such as simulating a real-world problem; function approximation, replacing the objective functions with cheaper functions; and evolutionary approximations, reducing the number of function evaluations by approximating similar individuals [28].

### III. MULTI-OBJECTIVE FISH BREEDING PROGRAM DESIGN

In this section, we will describe the MO-FBPD problem we formulate and investigate in this study. We will first describe the overall fish breeding simulation process, which was implemented via the SLiM software [10]. Then, we describe the decision variables and objective functions considered in the MO-FBPD problem.

#### A. Fish Breeding Simulation

We use the SLiM simulation software to simulate the fish breeding and mating process. The chromosome of each fish is represented as a string, where each bit in the string can take three possible values:  $B$ ,  $N$  and  $E$ . The letter “ $B$ ” means that the bit has been through a *beneficial* mutation. The letter “ $N$ ” indicates that the bit has been *neutrally* mutated, while the letter “ $E$ ” means the bit has not been mutated yet (empty mutation). Here, a mutation can occur with a low probability when a fish is born from its parents. A beneficial mutation may increase the quality of the fish (e.g., size and taste). On the other hand, a neutral mutation changes the genome of the fish without affecting its quality. However, it increases the diversity of the fish population, and may increase the potential of generating better fish in the future. Specifically, a fish chromosome  $\mathbf{C}_i$  can be represented as

$$\mathbf{C}_i = [c_{i1}, c_{i2}, \dots, c_{iK}], \quad (3)$$

where  $K$  is the chromosome length, and  $c_{ik} \in \{B, N, E\}$ . From the chromosome of a fish, we can calculate the number of beneficial and neutral mutations as follows:

$$\beta_{\text{ben}}(\mathbf{C}_i) = \sum_{k=1}^K \mathbb{I}(c_{ik} = B), \quad (4)$$

$$\beta_{\text{neu}}(\mathbf{C}_i) = \sum_{k=1}^K \mathbb{I}(c_{ik} = N), \quad (5)$$

where  $\mathbb{I}(X) = 1$  if the statement  $X$  is true, and 0 otherwise.

Algorithm 1 shows the details of the SLiM simulation process. Given an initial fish population  $\mathcal{P}_{\text{init}}$ , the selection

strength parameter  $\theta$ , the number of generations for the breeding  $G_{\text{breed}}$ , and the random seed  $\text{seed}$  for the parent selection and mating, the simulation will return the final fish population  $\mathcal{P}_{G_{\text{breed}}}$ .

In each breeding generation, the top  $N \times (1 - \theta)$  individuals from the current populations are first selected based on the selection strength, and placed into the mating pool. Then, a new offspring population is generated from the individuals in the mating pool. Specifically, we repetitively select two parents  $p_1$  and  $p_2$  from the mating pool using roulette wheel selection (proportional to  $\beta_{\text{ben}}$ ), and generate two offspring  $o_1$  and  $o_2$  through the  $\text{mating}(\cdot)$  function.

---

#### Algorithm 1: FishBreeding( $\mathcal{P}_{\text{init}}, \theta, G_{\text{breed}}$ )

---

**Input:**  $\mathcal{P}_{\text{init}}$ : the initial fish population  
**Input:**  $\theta$ : the selection strength  
**Input:**  $G_{\text{breed}}$ : the number of generations for breeding  
**Input:**  $\text{seed}$ : the random seed  
**Output:**  $\mathcal{P}_{\text{bred}}$ : the fish population after the breeding process

```

1  $N = |\mathcal{P}_{\text{init}}|$ ; // fish population size
2  $\mathcal{P}_0 = \mathcal{P}_{\text{init}}, g = 0$ ;
3 while  $g < G_{\text{breed}}$  do
  // fish mating for one generation
4    $\text{pool} = \text{top } N \times (1 - \theta) \text{ individuals in } \mathcal{P}_g$ ;
5    $\mathcal{P}_{g+1} = \emptyset$ ;
6   while  $|\mathcal{P}_{g+1}| < N$  do
     // born new fish from the mating pool
7      $p_1, p_2 = \text{roulette wheel selection from } \text{pool}$ ;
8      $o_1, o_2 = \text{mating}(p_1, p_2)$ ;
9      $\mathcal{P}_{g+1} = \mathcal{P}_{g+1} \cup \{o_1, o_2\}$ ;
10  end
11   $g = g + 1$ ;
12 end
13 return  $\mathcal{P}_g$ ;

```

---

Algorithm 2 shows the mating process of two parents to produce two offspring. First, a few cut points are randomly selected from the chromosome, and the substrings between the cut points are swapped between the two parents. After that, each gene of each chromosome will have a tiny probability to be mutated, i.e. its value will be randomly re-sampled from  $\{B, N\}$  if it currently is  $E$ .

---

#### Algorithm 2: Mating( $p_1, p_2$ )

---

**Input:**  $p_1$  and  $p_2$ : the two parents  
**Output:**  $o_1$  and  $o_2$ : the two offspring

```

1 Randomly choose a number of cut points in the chromosome;
2 Swap substrings of  $p_1$  and  $p_2$  at the cut points to get  $o_1$  and  $o_2$ ;
3 foreach  $c_k \in o_1$  and  $o_2$  do
4   if  $c_k = E$  then
5     Re-sample  $c_k$  from  $\{B, N\}$  with a tiny mutation
     probability;
6   end
7 end
8 return  $o_1, o_2$ ;

```

---

#### B. Decision Variables and Domain

In this study, we consider only a single decision variable, which is the selection strength  $\theta$  that determines the top percentage of the fish population to be selected into the mating pool. The selection strength ranges from 0 to  $1 - 2/N$ , where

$N$  is the fish population size. If the selection strength is 0, then all the fish in the current population can be selected for mating regardless of their quality. On the other hand, if the selection strength is  $1 - 2/N$ , then only the top 2 fish individuals (with the largest number of beneficial mutations) in the population will be selected for mating.

As a result, a (simple) fish breeding program is represented as a single number  $\theta \in [0, 1 - 2/N]$ , which is the selection strength in each breeding generation.

It is noteworthy that from line 4 of Algorithm 1, there are essentially  $N - 1$  unique possible situations for the selection strength (i.e., from top 2 to the entire fish population). In other words, the search space seems to be linear to the problem size. However, exhaustive search is usually not possible in the real world due to two reasons: (1) the fish population size can be very large in the real world (e.g., millions of fish in the population); (2) the fish population size can vary from one scenario to another. To generalise among the scenarios with different fish population sizes, the selection strength need to be kept as a continuous percentage value rather than a discrete number, which cannot be enumerated.

### C. Objective Functions

In this study, we consider to maximise the following two metrics to evaluate the quality of a fish breeding program.

- The *average quality of fish* in the final population after the breeding process.
- The *diversity* of the final population after the breeding process.

Obviously, both metrics are based on the final population after the breeding process. Given a final fish population  $\mathcal{P}$  obtained by a fish breeding program  $\theta$ , the above two metrics of  $\theta$  can be calculated as follows.

$$\max f_1(\theta; \mathcal{P}) = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{C}_i \in \mathcal{P}} \beta_{\text{ben}}(\mathbf{C}_i), \quad (6)$$

$$\max f_2(\theta; \mathcal{P}) = \frac{1}{|\mathcal{P}|} \sum_{k=1}^K \mathbb{I}(\exists \mathbf{C}_i \in \mathcal{P}, c_{ik} = N), \quad (7)$$

Eq. (6) indicates the average quality of fish in the final population, which is reflected by the number of beneficial mutations in each fish. Eq. (7) shows the number of genes in the chromosome that have been mutated neutrally, divided by the number of fish. For example, if there are three fish, whose chromosomes are “NNEEE”, “NEEEE” and “EENEE”, then there are three genes (in the first three positions) with neutral mutation, and the diversity of the population is  $3/3 = 1$ .

Note that the above metrics depends on the SLiM simulation, which is random (in initial population and mating process). That is, even with the same breeding program, different simulations (i.e., random seeds) will produce different final fish population, and different values of Eqs. (6) and (7).

As the ultimate goal is to obtain a high-quality fish breeding program on an unseen fish population (i.e., generalisation), we use a set of training simulations to evaluate the breeding

programs. Each training simulation contains the initial fish population, number of breeding generations, and random seed for the breeding process.

Algorithm 3 shows the calculation of the two objectives of a breeding program  $\theta$ . It runs each training simulation with the selection strength  $\theta$  (note that  $\text{FishBreed}(\theta; \text{sim})$  in line 3 is essentially the same as Algorithm 1 by taking the corresponding parameters from the simulation configuration). Then, for each bred fish population, it calculates the two metrics based on Eqs. (6) and (7). Finally, the objective values are calculated by averaging over all the training simulations.

---

#### Algorithm 3: Objectives( $\theta; \mathcal{S}_{\text{train}}$ )

---

**Input:**  $\theta$ : the selection strength  
**Input:**  $\mathcal{S}_{\text{train}}$ : the training simulations  
**Output:** the objective values of  $\theta$

```

1  $f_1 = 0, f_2 = 0;$ 
2 for  $\text{sim} \in \mathcal{S}_{\text{train}}$  do
   // run the simulation breeding process
3    $\mathcal{P}(\text{sim}) = \text{FishBreed}(\theta; \text{sim});$ 
4   Calculate  $f_1$  and  $f_2$  by Eqs. (6) and (7);
5    $f_1 = f_1 + f_1(\theta; \mathcal{P}(\text{sim}));$ 
6    $f_2 = f_2 + f_2(\theta; \mathcal{P}(\text{sim}));$ 
7 end
   // average over all training simulations
8  $f_1 = f_1/|\mathcal{S}_{\text{train}}|, f_2 = f_2/|\mathcal{S}_{\text{train}}|;$ 
9 return  $f_1, f_2;$ 

```

---

### D. Relationships between Objectives

In Eq. (7), we know that

$$\sum_{k=1}^K \mathbb{I}(\exists \mathbf{C}_i \in \mathcal{P}, c_{ik} = N) \leq \sum_{\mathbf{C}_i \in \mathcal{P}} \sum_{k=1}^K \mathbb{I}(c_{ik} = N), \quad (8)$$

and the equality is satisfied if all the individuals have their neutral mutations (“N”) in different genes.

From the definition of  $\beta_{\text{neu}}$  in Eq. (5), we have

$$\sum_{k=1}^K \mathbb{I}(\exists \mathbf{C}_i \in \mathcal{P}, c_{ik} = N) \leq \sum_{\mathbf{C}_i \in \mathcal{P}} \beta_{\text{neu}}(\mathbf{C}_i), \quad (9)$$

It is clear that for any fish chromosome  $\mathbf{C}_i$ , we have

$$\beta_{\text{ben}}(\mathbf{C}_i) + \beta_{\text{neu}}(\mathbf{C}_i) \leq K, \quad (10)$$

where  $K$  is the length of the chromosome. The equality is met when all the bits have mutated (either beneficial or neutral). Therefore, for any fish chromosome, if all the bits have mutated, we have  $\beta_{\text{ben}}(\mathbf{C}) = K - \beta_{\text{neu}}(\mathbf{C})$ . Given any population  $\mathcal{P}$ , from Eqs. (6) and (7) we can have

$$f_2(\theta; \mathcal{P}) \leq K - f_1(\theta; \mathcal{P}). \quad (11)$$

In addition, if all the genes of all the chromosomes are mutated and different chromosomes have their neutral mutations on different genes, then the two objectives should be on the line  $x + y = K$ . This obviously shows the conflicting relationships between the two objectives.

In practice, however, one cannot guarantee that the two sides of Eq. (11) are equal, and the objective values of the fish

chromosomes always on that line. This is because Eq. (11) assumes that every bit in the chromosome has mutated, which is not true in practice. On the contrary, the chromosome length is very long (e.g.,  $> 10000$ ) and the mutation probability is very tiny. As a result, only a small amount of bits will be mutated. Furthermore, one cannot guarantee that different chromosomes have no common genes with neutral mutations.

As a result, the Pareto front of this problem should be below the line  $x + y = K$ .

#### IV. NON-DOMINATED SORTING GENETIC ALGORITHM-II FOR MO-FBPD

In this study, we use the well-known Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [22] to investigate the MO-FBPD problem. The flowchart of the process is shown in Fig. 1. It uses a set of training simulations to obtain a set of non-dominated breeding programs, and then apply them to the unseen test simulations to see their test performance.

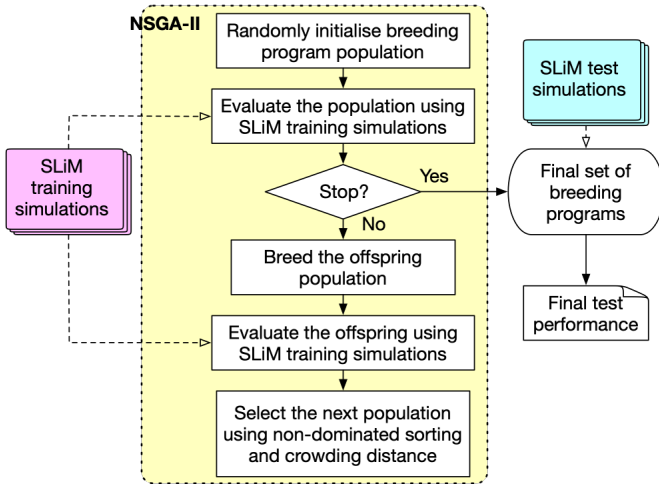


Fig. 1. The flowchart of the NSGA-II for MO-FBPD.

The pseudo code of NSGA-II training for MO-FBPD is shown in Algorithm 4. It follows the conventional NSGA-II framework. First, a population of breeding programs  $\Theta_0$  is initialised. Then, at each generation, an offspring population  $\Theta'$  with the same size as the current population is first generated by parent selection, crossover and mutation operators. After all the offspring have been generated, the current population and the offspring population are combined, and the next population is selected by the non-dominated sorting and crowding distance (i.e.,  $\text{NDSortingCD}(\cdot)$  in line 23). Finally, the first front of the final population is returned.

An individual is directed represented as the selection strength  $\theta$ . It is a continuous value taken from  $[0, 1 - 2/N]$ . The parents are selected by the binary tournament selection. That is, two individuals  $I_1$  and  $I_2$  are first randomly selected from the population. Then one individual is selected based on the following criteria.

- If  $\text{rank}(I_1) < \text{rank}(I_2)$ , then select  $I_1$ ;

#### Algorithm 4: NSGA-II training for MO-FBPD

---

**Input:**  $\mathcal{S}_{\text{train}}$ : the training simulations  
**Output:** the non-dominated set of breeding programs  $\Theta^*$

```

1  $g = 0$ ;
2 Randomly initialise a population of breeding programs  $\Theta_0$ ;
3 foreach  $\theta \in \Theta_0$  do
4   |  $f_1(\theta), f_2(\theta) = \text{Objectives}(\theta; \mathcal{S}_{\text{train}})$ ;
5 end
6 while stopping criteria are not met do
7    $\Theta' = \{\}$ ; // offspring population
8   while  $|\Theta'| < |\Theta_g|$  do
9     Select  $\theta_1, \theta_2 \in \Theta_g$  by binary tournament selection;
10    if  $\text{rand} < \text{Pr}_c$  then
11      | Generate two offspring  $\theta'_1, \theta'_2 = \text{SBX}(\theta_1, \theta_2, \eta_c)$ ;
12    else
13      |  $\theta'_1 = \theta_1, \theta'_2 = \theta_2$ ;
14    end
15    if  $\text{rand} < \text{Pr}_m$  then
16      |  $\theta'_1 = \text{PolyMutation}(\theta'_1, \eta_m)$ ;
17      |  $\theta'_2 = \text{PolyMutation}(\theta'_2, \eta_m)$ ;
18    end
19     $f_1(\theta'_1), f_2(\theta'_1) = \text{Objectives}(\theta'_1; \mathcal{S}_{\text{train}})$ ;
20     $f_1(\theta'_2), f_2(\theta'_2) = \text{Objectives}(\theta'_2; \mathcal{S}_{\text{train}})$ ;
21     $\Theta' = \Theta' \cup \{\theta'_1, \theta'_2\}$ ;
22  end
23   $\Theta_{g+1} = \text{NDSortingCD}(\Theta_g \cup \Theta')$ ;
24   $g = g + 1$ ;
25 end
26 return the first front of  $\Theta_g$ ;
```

---

- If  $\text{rank}(I_1) = \text{rank}(I_2)$  and  $CD(I_1) \geq CD(I_2)$ , then select  $I_1$ ;
- Otherwise, select  $I_2$ .

The Simulated binary crossover (SBX) operator and polynomial mutation (PolyMutation) operator are used for generating offspring. More details can be found from [29] and [30].

#### V. EXPERIMENTAL STUDIES

To investigate the performance of NSGA-II in solving MO-FBPD, we conduct the training and test process as shown in Fig. 1, and evaluate the training and test performance of the obtained breeding programs.

##### A. SLiM Simulation Configuration

We create 30 SLiM simulations for training, and another 30 SLiM simulations for each test case. In the training simulations, there are 500 fish in the population. In the test simulations, there are 1000 and 2500 fish in the population. We set a different number of fish in the population in the training and test simulations, to assess the generalisation of the obtained non-dominated breeding programs among different fish population sizes. In both the training and test simulations, the breeding lasts for 50 generations. The length of chromosome is set to 9999.

Note that the SLiM simulations starts with fish with no mutation (i.e., chromosome bits are all “E”). To avoid this initial bias, we pre-run the simulation for 500 breeding generations (zero selection strength) to obtain an initial population with diverse fish chromosomes.

## B. NSGA-II Parameter Settings

Table I gives the parameter settings of NSGA-II. In the experiments, NSGA-II was run 30 times independently, to get 30 sets of non-dominated breeding programs. Then, each set of breeding programs is tested on the test simulations (1000 fish and 2500 fish), respectively.

TABLE I  
THE NSGA-II PARAMETER SETTINGS.

Parameter	Value
Population size	100
Number of generations	250
Crossover probability $Pr_c$	0.9
Mutation probability $Pr_m$	0.01
Crossover parameter $\eta_c$	2
Mutation parameter $\eta_m$	50

For comparison, for each simulation scenario (500, 1000 and 2500 fish population size), we run the enumeration method (selecting the top 2 fish, 3 fish, . . . , the whole population) to obtain the optimal Pareto front of the breeding programs. The results are shown under “Enumeration (Optimal)”.

We use the hyper-volume (HV) and inverted generation distance (IGD) as the performance metric to evaluate the evolved Pareto fronts. First, we normalise the two objective values into the range [0, 1]. Then, for the HV calculation, we select the reference point of 0.0. For the IGD, the optimal Pareto front obtained by the enumeration method is used for the calculation.

## C. Results and Discussions

Table II presents the mean and standard deviation of the HV values of the enumeration (optimal) and NSGA-II on the training dataset and the two test datasets (1000 and 2500 fish, respectively). The enumeration method is deterministic, therefore it returns a single value. We conduct the Wilcoxon rank sum test with significance level of 0.05 to compare between the results of NSGA-II and the optimal results obtained by enumeration. The significantly better results are highlighted in bold.

TABLE II  
THE HV RESULTS OF NSGA-II AND ENUMERATION (OPTIMAL).

Scenario	Enumeration (Optimal)	NSGA-II mean (std)
Training	<b>0.577441</b>	0.575876 (0.000225)
Test (1000 fish)	<b>0.553054</b>	0.530340 (0.001104)
Test (2500 fish)	<b>0.500687</b>	0.487704 (0.001337)

From Table II we can see that the training HV of NSGA-II is very close to the optimum (0.576 vs 0.577), although it is statistically significantly worse. For the two test cases with 1000 and 2500 fish, we can see that the generalised HV value of NSGA-II is also very close to the optimal HV. This is a good sign, which shows that the breeding programs obtained

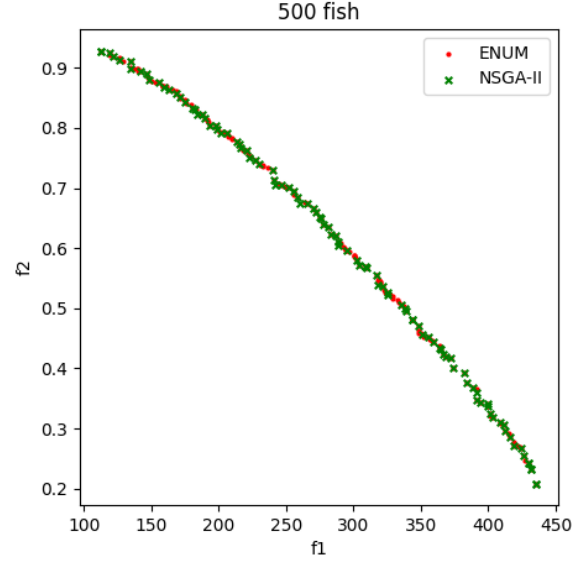


Fig. 2. The distribution of the objective values obtained by NSGA-II and enumeration on the **training** dataset (500 fish).

by NSGA-II for 500 fish can be generalised to 1000 and 2500 fish breeding processes very well.

Table III shows the IGD of NSGA-II on the training and test datasets. Note that the enumeration results are omitted here, since they are optimal, and can always achieve IGD of zero.

TABLE III  
THE IGD MEAN AND STANDARD DEVIATION OF NSGA-II IN 30 RUNS.

Scenario	NSGA-II IGD
Training	0.003447 (0.000287)
Test (1000 fish)	0.013498 (0.000647)
Test (2500 fish)	0.007991 (0.000446)

From Table III, we can see that NSGA-II can achieve very small IGD values on the training and test scenarios. This is consistent with the HV results.

Figs. 2–4 show the distributions of the objective values obtained by NSGA-II and enumeration (optimal) on the training and test datasets. From Fig. 2, we can see that the Pareto front obtained by NSGA-II can cover the true Pareto front very well. The green points are almost overlapping with the red ones, including the extreme points, and missed only a few regions in the middle. It can also be seen that  $f_2$  is much smaller than  $f_1$ . This is because different fish may have neutral mutation in the same gene, which does not increase  $f_2$ .

From Figs. 3 and 4, we can see that the results of NSGA-II are still mostly overlapping with the true Pareto front. The slightly worse test HV and IGD values of NSGA-II were mainly due to the weaker coverage, specifically in the region with large  $f_1$ . In other words, the breeding programs with large

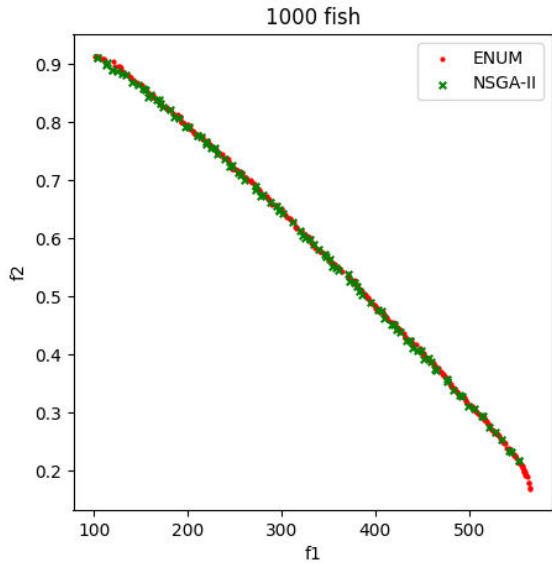


Fig. 3. The distribution of the objective values obtained by NSGA-II and enumeration on the **test** dataset (**1000** fish).

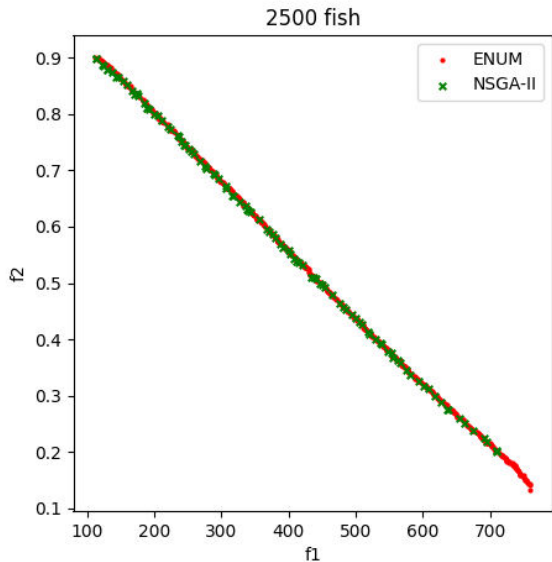


Fig. 4. The distribution of the objective values obtained by NSGA-II and enumeration on the **test** dataset (**2500** fish).

$f_1$  (number of beneficial mutations) are harder to generalise to different number of breeding fish.

From the figures, it is also interesting to see that the Pareto fronts are almost aligned with a straight line. That is, the relationship between the two objectives is linear. It is worth noting that in all the figures, the Pareto front is far below the upper bound  $f_1 + f_2 = K$  ( $K = 9999$ ). This indicates that there are still many genes in the fish chromosome without any mutation.

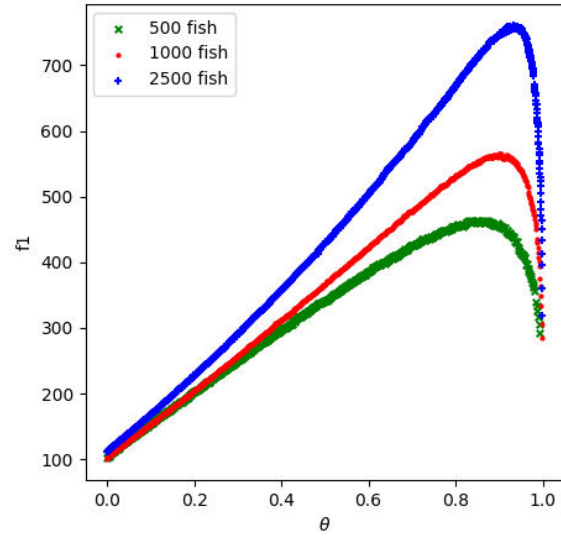


Fig. 5. Relationship between the selection strength  $\theta$  and  $f_1$ .

Figs. 5 and 6 show the relationship between the selection strength ( $\theta$ ) and the two objectives for the scenarios with 500, 1000 and 2500 fish. The results were obtained by enumeration. From Fig. 5, we can see that for all the scenarios, as  $\theta$  increases,  $f_1$  (number of beneficial mutations) tends to increase monotonically until  $\theta = 0.8680$ . The monotonic increasing pattern of  $f_1$  is intuitive, as the parent fish in the mating pool tend to have more beneficial mutations as  $\theta$  increases. It is interesting to see that when  $\theta$  is too large, the number of beneficial mutations tends to decrease, due to the lack of diversity in the population.

Fig. 6 shows that as  $\theta$  increases,  $f_2$  (the number of neutral mutations) tends to decrease. This is consistent with the increase of the beneficial mutations, since the two numbers tend to conflict with each other.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, we investigated the MO-FBPD problem that has important applications in the aquaculture industry. The problem aims to optimise both the short-term gain (e.g., beneficial mutations in the fish gene) and long-term gain (e.g., genetic diversity in the fish population). We developed a SLiM-based simulator for the fish breeding process, and formulate the MO-FBPD problem, including the decision variables and objective functions. Then, we applied NSGA-II to the problem and analysed its performance on different scenarios. The experimental results showed that the breeding programs (represented by the selection strength, i.e., percentage of top fish to be mated) obtained by NSGA-II are able to generalise well through different fish population size. This is an important observation, since it implies that the breeding program design is generic, and can be applied to any future breeding process regardless of its fish population size. In



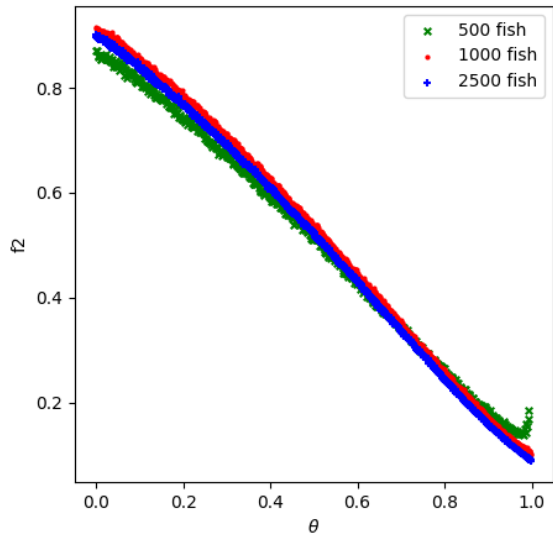


Fig. 6. Relationship between the selection strength  $\theta$  and  $f_2$ .

addition, we have observed that it is undesirable to select the mating fish too greedily, as it will behave oppositely as we expected (decrease the number of beneficial mutations and increase the number of neutral mutations).

In future, we plan to further improve the simulation process to make it closer to reality by including more real-world factors. We will also improve the definition of objective functions to include more considerations (e.g., multiple traits). We will also develop better MOEAs to improve the optimisation results.

## REFERENCES

- [1] L. Miguel Antonio and C. A. Coello Coello, "Coevolutionary multi-objective evolutionary algorithms: Survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 6, pp. 851–865, 2018.
- [2] N. Jozefowicz, F. Semet, and E.-G. Talbi, "Enhancements of nsga ii and its application to the vehicle routing problem with route balancing," in *International Conference on Artificial Evolution (Evolution Artificielle)*. Springer, 2005, pp. 131–142.
- [3] S. Wang, D. Zhao, J. Yuan, H. Li, and Y. Gao, "Application of nsga-ii algorithm for fault diagnosis in power system," *Electric Power Systems Research*, vol. 175, p. 105893, 2019.
- [4] M. Atiquzzaman, S.-Y. Liong, and X. Yu, "Alternative decision making in water distribution network with nsga-ii," *Journal of water resources planning and management*, vol. 132, no. 2, pp. 122–126, 2006.
- [5] A. Kamjoo, A. Maheri, A. M. Dizqah, and G. A. Putrus, "Multi-objective design under uncertainties of hybrid renewable energy system using nsga-ii and chance constrained programming," *International journal of electrical power & energy systems*, vol. 74, pp. 187–194, 2016.
- [6] S. Honda, T. Igarashi, and Y. Narita, "Multi-objective optimization of curvilinear fiber shapes for laminated composite plates by using nsga-ii," *Composites Part B: Engineering*, vol. 45, no. 1, pp. 1071–1078, 2013.
- [7] S. Torabi, S. Alibabaei, B. B. Bonab, M. Sadeghi, and G. Faraji, "Design and optimization of turbine blade preform forging using rsm and nsga ii," *Journal of Intelligent Manufacturing*, vol. 28, no. 6, pp. 1409–1419, 2017.
- [8] D. Akdemir and J. I. Sánchez, "Multi-objective optimized breeding strategies," *bioRxiv*, p. 209080, 2017.
- [9] D. Akdemir, W. Beavis, R. Fritsche-Neto, A. K. Singh, and J. Isidro-Sánchez, "Multi-objective optimized genomic breeding strategies for sustainable food improvement," *Heredity*, vol. 122, no. 5, pp. 672–683, 2019.
- [10] B. C. Haller and P. W. Messer, "SLiM 3: Forward Genetic Simulations Beyond the Wright–Fisher Model," *Molecular Biology and Evolution*, vol. 36, no. 3, pp. 632–637, 01 2019.
- [11] M. Besson, H. Komen, G. Rose, and M. Vandeputte, "The genetic correlation between feed conversion ratio and growth rate affects the design of a breeding program for more sustainable fish production," *Genetics Selection Evolution*, vol. 52, no. 1, pp. 1–10, 2020.
- [12] M. Holtmark, A. K. Sonesson, B. Gjerde, and G. Klemetsdal, "Number of contributing subpopulations and mating design in the base population when establishing a selective breeding program for fish," *Aquaculture*, vol. 258, no. 1-4, pp. 241–249, 2006.
- [13] T. Meuwissen, "Maximizing the response of selection with a predefined rate of inbreeding," *Journal of Animal Science*, vol. 75, no. 4, pp. 934–940, 04 1997. [Online]. Available: <https://doi.org/10.2527/1997.754934x>
- [14] M. Dupont-Nivet, M. Vandeputte, P. Haffray, and B. Chevassus, "Effect of different mating designs on inbreeding, genetic variance and response to selection when applying individual selection in fish breeding programs," *Aquaculture*, vol. 252, no. 2-4, pp. 161–170, 2006.
- [15] A. Abraham and L. Jain, "Evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization*. Springer, 2005, pp. 1–6.
- [16] C. C. Coello, "Evolutionary multi-objective optimization: a historical view of the field," *IEEE computational intelligence magazine*, vol. 1, no. 1, pp. 28–36, 2006.
- [17] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and evolutionary computation*, vol. 1, no. 1, pp. 32–49, 2011.
- [18] E. Zitzler, M. Laumanns, and S. Bleuler, "A tutorial on evolutionary multiobjective optimization," *Metaheuristics for multiobjective optimization*, pp. 3–37, 2004.
- [19] S. Jiang, J. Zhang, Y.-S. Ong, A. N. Zhang, and P. S. Tan, "A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2202–2213, 2014.
- [20] H. Ishibuchi, N. Tsukamoto, Y. Sakane, and Y. Nojima, "Hypervolume approximation using achievement scalarizing functions for evolutionary many-objective optimization," in *2009 IEEE Congress on Evolutionary Computation*. IEEE, 2009, pp. 530–537.
- [21] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *International conference on parallel problem solving from nature*. Springer, 2004, pp. 832–842.
- [22] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [23] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," *TIK-report*, vol. 103, 2001.
- [24] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [25] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, "A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms," *Soft Computing*, vol. 23, no. 9, pp. 3137–3166, 2019.
- [26] Z. Lv, L. Wang, Z. Han, J. Zhao, and W. Wang, "Surrogate-assisted particle swarm optimization algorithm with pareto active learning for expensive multi-objective optimization," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 3, pp. 838–849, 2019.
- [27] Q. Zhou, J. Wu, T. Xue, and P. Jin, "A two-stage adaptive multi-fidelity surrogate model-assisted multi-objective genetic algorithm for computationally expensive problems," *Engineering with Computers*, Aug 2019.
- [28] L. V. Santana-Quintero, A. A. Montano, and C. A. C. Coello, "A review of techniques for handling expensive functions in evolutionary multi-objective optimization," *Computational intelligence in expensive optimization problems*, pp. 29–59, 2010.
- [29] K. Deb, R. B. Agrawal *et al.*, "Simulated binary crossover for continuous search space," *Complex systems*, vol. 9, no. 2, pp. 115–148, 1995.
- [30] K. Deb and H.-G. Beyer, "Self-adaptive genetic algorithms with simulated binary crossover," *Evolutionary computation*, vol. 9, no. 2, pp. 197–221, 2001.