Simulating and Evolving Fish Schools in Settings with Predators and Food

Huvar Ondřej, Kalabis Tomáš, Tufchi Ariyan

1 Introduction

Several oceanic fishes, similar to other animals, present social behaviour. The primary purpose of this phenomenon is to increase mutual survivability, which can be subdivided into (i) mutual protection and (ii) collaborative completion of additional tasks. It is a well-known fact that many fish species live their entire lives in schools despite the constraints on swimming movements and competition in food-scarce areas. However, grouping fishes together is a fact, and its benefits far outweigh the drawbacks [1, p.261]. We can explore and study the intricate dynamics of fish swarm behaviour by recreating these real-world interactions in a simulation. Particularly, approaches based on Swarm Intelligence (SI) proved suitable [2, 3]. The adoption of SI enables the simulation of fish swarm behaviour by enacting interactions and movements of individual fish inside the swarm. Furthermore, the addition of a predator, in this case, a shark, as well as the availability of food, enhances the simulated environment by recreating the predator-prey interaction and natural resource availability that exist in real-world settings.

In nature, the emergent behaviour of fish schools is very complex, as it evolved over millennia via natural selection. Similarly, the behaviour of artificially modelled fish is also complex, depending on various parameters. We thus employ an approach based on Evolutionary Algorithm (EA) to optimize the parameters governing fish swarm behaviour. This approach utilizes natural selection and evolution concepts by refining parameters over subsequent generations, enhancing the performance and efficacy of the fish swarm in accomplishing its survival. Incorporating EA also allows us to identify different (sub-)optimal parameterizations representing various strategies that fish may potentially employ.

The objectives of this study are: Explore and gain insight into complex behavioural patterns that emerge as cause of fish grouping in schools and trying to avoid predators. Analyze how these patterns change when fish have to prioritize between survival and food consumption. Examine the robustness of such behavioural strategies. Furthermore, as we seek to optimize the swarm parameters using an evolutionary algorithm, we plan to assess if various settings of the evolution influence the resulting strategies.

1.1 Related work

Underlying mechanisms of fish behaviour and schooling principles are described in depth in [4]. Some aspects of shark's behaviour are captured in [5, 6]. In [2, 3], the authors describe the Artificial Fish Swarm Algorithm and its applications. This is mainly an optimization method that (i) does not try to model reality too much and (ii) does not involve predators. The comparison of the Predator&Prey model and Particle Swarm Optimization is discussed in [7]. Although it focuses on different models, it gives insight into predator modelling. Lastly, the process of combining SI and EA approaches, although of different nature than ours, is described in [8].

2 Method

2.1 Simulation

In order to model and simulate the scenario, we use an approach based on Swarm Intelligence. Particularly, our algorithm is a modification and extension of the Boid algorithm [9]. We extend the original algorithm in several ways and try to make some aspects of the simulation biologically relevant (such as the "forces" affecting the fish).

Our simulation is set in a 2D space on a square grid. However, the grid's borders are wrapped, essentially creating an "infinite" grid. In the beginning, all entities are randomly placed on the grid. In every simulation step, we compute the state of each shark, fish and food piece. The objective of a shark is to hunt fish. The objective of a fish is to stay alive and eat food (if specified in the objective function). Their movement is affected by each other through the use of specific "forces", which are represented as vectors (with a direction and magnitude). Several values parametrize our simulation. We distinguish between two types of parameters – *fixed* and *evolvable*. The fixed parameters have the same values for each simulation we run, only changing for the purpose of some experiments (for instance, to assess the robustness). The evolvable parameters are the ones to be evolved by the EA. They are solely the multiplicative constants of each force affecting fish. The forces, the fixed parameters and the whole mechanism are described in the following subsections.

2.1.1 Modelling Fish Schools

Each fish is represented as an ellipse and is affected by several forces. After thorough research, we decided to model fish behaviour by implementing the following forces: *cohesion*, *alignment*, *separation*, *momentum*, *shark repulsion*, and *food attraction*.

In [2], the authors describe three kinds of forces used for fish school modelling – *swarm* behaviour, *follow* behaviour, and *jamming* behaviour. These are essentially the same as *cohesion*, *alignment*, and *separation*, respectively, which are the forces used in the original Boid algorithm [9]. These three principles are also used in [3], where they are called *cohesion*, *unification*, and *compartmentation*, respectively. For these reasons, we also implement them for our simulation. In [4], the authors describe how information flow between the fish is relatively short-range, affected by the limits of fish senses. We incorporate that into our simulation, limiting the fish *sense distance* (modelled as a circle around the fish). Fish is thus only affected by its neighbours up to a certain radius ("local fish neighbourhood"). Therefore, the cohesion steers to move towards the average position (centre of mass) of fish in a local neighbourhood, the alignment steers towards the average heading of fish in a neighbourhood and the separation steers to avoid overcrowded local areas.

Additionally, each fish is affected by a *momentum* force, as advocated in [10]. This force prevents a sudden change of direction. With the addition of predators (sharks), we also introduce the *shark repulsion* force which repulses a fish from a shark whenever the shark is in the distance of the fish's senses. This repulsive force preserves for several steps, even though the shark is no longer in the fish's neighbourhood. This corresponds to a fear reaction of the fish, which is also used by the authors of [7]. Finally, with the addition of food, we also add *food attraction* force, attracting the fish to the closest food in the neighbouring area.

The composed force vector (defining the fish's direction and step length) is calculated by first summing up the normalized vectors of each force multiplied by a corresponding constant and then adding a slight random noise. When there are no entities in the fish's neighbourhood, this noise corresponds to the *searching* behaviour of [2] or *free move* behaviour of [10]. Further, we enforce the *maximal speed limit*, which prevents any fish from reaching unnatural speed. This is implemented as a maximal magnitude of the overall force vector, as in [10]. Lastly, there is one more repulsion between any two fish whose physical dimensions would overlap.

2.1.2 Modelling Sharks

Our simplistic artificial sharks are also represented as an ellipse. The goal of a shark is to eat fish. That happens when the centre of a fish gets inside the shark's "mouth", represented as a small circle at the front of the shark (see the red circle in Figure 1). Many shark species are known to prefer to use vision while hunting prey located close to them [5]. We thus simplify our sharks to rely on this sense. Similarly to our fish, the artificial shark only senses neighbouring fish up to a given *sense distance*. Furthermore, as advocated in [6], a part of this area behind the shark is excluded and is referred to as a *visual blind spot* (see shark sense area in green in Figure 1). Particularly, we use a 40° blind angle [6].

The artificial shark has two modes of behaviour. When there are no neighbouring fish, *searching*

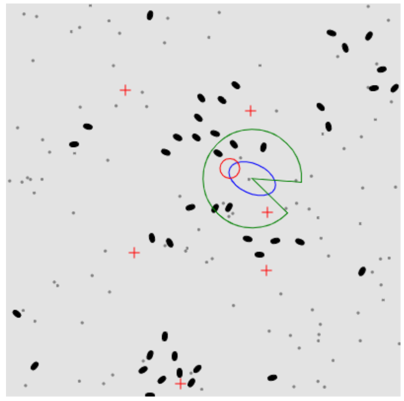


Figure 1: The screenshot captures one step of the visualization of a simulation. The shark is the blue ellipse with its *kill radius* as a red circle and its *sense distance* as a green circle (with the blind spot at the back). The black ellipses and grey dots denote fish and food, respectively. Each red cross denotes the place where a fish was eaten.

mode is activated – sharks randomly adjust their direction to search for prey. When a shark senses fish nearby, *hunt* mode is activated. The *hunt direction* is computed as the direction to the centroid of visible fish. The resulting force vector affecting the shark is computed by combining the hunt or search vector with a shark's *momentum* vector. This way, the shark's direction does not change drastically. Finally, same as for the fish, it is ensured that the shark's speed does not exceed a certain limit.

2.1.3 Modelling Food

The food pieces are randomly drifting in the grid (in little steps), and they are not affected by any external force. When a fish eats a piece of food, a new piece is randomly placed in the grid. This way, there is the same amount of food every step.

2.1.4 Visualization

Additionally, we have created a visualization tool for simulations. We have utilized the *p5.js* JavaScript library. The visualization captures the movement of sharks, fish, and food. It also captures the *kill radius* and the *sense distance* (excluding shark visual blind spot) of each shark. Lastly, there is a tombstone at every location where a fish dies. An example with a description can be seen in Figure 1.

2.2 Evolution

In order to optimize the parameters of the artificial fish, we utilize an approach based on Evolutionary Algorithms (EA). The parameters to optimize are the "strengths" of the individual forces that affect the fish (as described in Section 2.1). They are constants used to multiply each vector representing the forces (in the simulation). Each individual in the evolution's population thus stands for one simulation setting and is represented by a list of several real numbers. We do not force any external limitation on the evolvable parameters, except for the parameter regarding *momentum* force. This parameter only takes a value from [0.3,1] to be more realistic (e.g., to prevent large and quick changes in fish speed and direction). Our general approach follows the classical EA template described in [11]. For completeness, the high-level pseudocode is given as Algorithm 1 in the Appendix.

Objective function Each individual in our population represents a parametrisation for a simulation. Due to the stochastic nature of the simulation, it is a complex task to evaluate its fitness. Therefore, to achieve robust results, we re-run each simulation six times [1]. We take the average number of fish eaten by a shark as a base evaluation score. For some of the experiments, we also consider the number of food pieces eaten, as described in Section 3.1.

[1] We chose the number 6 particularly to allow for smooth parallelization on standard laptops.

Initial population The way how we generate the initial population proved to be an important aspect. As parameters are essentially used as multiplicative constants in the simulation, we need both values larger than 1 (for multiplication) and between zero and one (for division). When generating each parameter, we uniformly sample from either interval [1, 20] or [0, 1]. The choice of the sampling interval for each parameter is random. Value 20 was chosen based on our experiments with the simulation.

Selection Strategy We select the individuals for reproduction (crossover or mutations) using a tournament strategy. By repeatedly arranging tournaments and taking the winners, we sample the same number of individuals as was in the original population (only this time with repetitions). We also experimented with stochastic universal sampling. However, it did not affect the performance.

Reproduction Strategy As individuals consist of a list of real numbers, we devised a strategy inspired by two well-known methods, Evolutionary Strategies (ES) and Genetic Algorithms (GA). Motivated by the ES approach, we heavily focus on mutations. We also apply a crossover operation (as used in GA) on a smaller scale. With a given probability, we first use a uniform crossover technique for offspring generation. This enables the parents to exchange values corresponding to the same parameter. Other widely used techniques, such as point crossover, are unsuitable for our settings (because they exchange values corresponding to different genes). We then generate several copies of both parents and offspring and apply a mutation operation with a designed probability to each individual's genes. By using several copies of each individual, we get differently mutated versions of each of them. For the mutations, we consider each parameter individually. To mutate a parameter, we randomly sample the number from [0.5,1] and randomly choose if to multiply or divide the current parameter with the selected value. This way, the parameter can get slightly larger or smaller (but not more than two times in a single step). Finally, we discard repetitive individuals from our population (e.g., when several copies of an individual did not mutate). We provide high-level pseudocode for the whole strategy as Algorithm 2 in the Appendix for completeness.

Replacement Strategy We employ elitist replacement, taking the best-performing individuals of the combined population. Before selecting the new generation, we filter out individuals that are potentially present in the combined population multiple times.

3 Design of Experiments

We design our experiments so that we can potentially observe various "efficient" modes of behaviour of the fish schools. At the same time, we make an effort to make our experiments robust. We focus on three main areas of interest – (i) experimenting with including food consumption in the objective function; (ii) assessing the robustness of the results via experimenting with the parameters of the simulation, mainly regarding sharks; (iii) assessing the robustness of the evolution process by experimenting with selection pressure and diversification vs intensification during the evolution.

There are many significant parameters of our simulation. We present the default values used in our experiments in Table 1. These settings are the same for all experiments unless stated otherwise.

Similarly, we present the list of parameter values for our baseline evolution in Table 2. The baseline evolution is designed to allow evolving into different settings (so we can explore them) and not get stuck in the same local minima each time. Some values are based on our experiments (e.g., a maximal number of 20 generations usually resulted in convergence), while others are designed to achieve specific goals. For instance, we set the mutation probability μ to 0.2 to achieve the expected mutation of at least one gene for an individual (there are six genes per individual). We vary the evolution's parameter values in some of our experiments to explore different settings. More on that in the following subsections.

To account for the inherent stochasticity of the evolution process, we repeat each experiment several times. We repeat the baseline evolution runs five times to properly explore various local minima ("types of behaviour") found by the evolution process. We re-run the other experiments three times. Moreover, we perform three versions of each experimental run, varying by the number of fish used in the underlying simulation. We use simulations with 100, 200 and 400 fish. With the

Parameter	**Value**
Grid size	400×400
Simulation steps	1000
Number of sharks	1
Number of food pieces	50
Fish ellipse dimensions	5×9
Shark ellipse dimensions	30×50
Parameter	**Value**
Fish max speed	4
Shark max speed	7
Fish sense distance	25
Shark sense distance	100
Shark kill radius	10
Shark blind angle	40°

Table 1: A list of parameter values used for our baseline simulation.

Parameter	**Abbr.**	**Value**
Mutation probability	μ	0.2
Crossover probability	κ	0.6
Mutation copies	m_c	3
Tournament size	θ	7
Parameter	**Abbr.**	**Value**
Population size	N	20
Maximal n. of generations	G_{max}	20
Early stopping generations	E	10
Simulations per individual	S	6

Table 2: A list of default parameter values used for our baseline evolution.

simulation's scene size, settings with more than 400 fish result in overcrowding, while no interesting patterns usually emerge with less than 100 fish. Together, we get 15 evolution runs for the baseline setting and nine for each experimental setting.

3.1 Objective Function and Food

We experiment with modifying the objective function of the evolution to account for food consumption. This allows us to explore how the importance of food (in relation to survival) affects fish behaviour and what types of patterns emerge. We extend the evaluation function in the following manner:

$$Eval = N_{\text{DEAD\_FISH}} - food\_weight \times N_{\text{FOOD\_EATEN}}$$

By varying the *food_weight* parameter, we can achieve different modes of behaviour. We thus experiment with this parameter. It is important to select an appropriate range of values to balance food consumption and survival. During our initial experiments (with 400 fish), fish usually have no problems consuming several thousand pieces of food during a simulation. We thus experiment with relatively small values ranging from $food\_weight = 0.0001$ (low food importance) to $food\_weight = 0.01$ (high food importance). It is expected that the values of the objective function may become negative.

3.2 Robustness assessment

3.2.1 Shark Parameters

First, we experiment with varying two parameters of the simulation. Specifically, we focus on the *number of sharks* and the *speed of a shark*. These experiments can help us better assess the robustness of the approach and solutions. Mainly, we are interested in whether the same strategies employed by fish could generalize for various settings. For both experiments, we are also interested in whether the objective function values (number of fish eaten) will show a linear trend (to the number of sharks or the shark's speed).

We explore the settings with up to three sharks. Experimenting with more than three sharks would be too much for the size of the fish population we employ. We also experiment with the parameter that limits the maximal *speed of a shark*. This parameter has a value of 7 in the baseline, being nearly twice as high as the maximal speed of fish (4). We experiment with values 5 and 9. Thus, the ratio between shark and fish speeds is modified from 1.75 to 1.25 and 2.25, respectively.

3.2.2 Selection Pressure

We designed our next round of experiments to investigate the impact of *mutation probability* and *tournament size* on the evolution results and performance of the successful fish populations. By manipulating these factors, we explore the trade-off between exploration and exploitation, as well as the balance between diversity and selection pressure within the search space. The mutation probability controls the level of exploration. Higher probabilities enable wider solution exploration, aiding in escaping local optima but potentially leading to slower convergence. Lower probabilities encourage the exploitation of current solutions, promoting faster convergence but increasing the risk of getting stuck in sub-optimal solutions. Varying tournament sizes allows us to examine the diversity-selection pressure relationship. Larger sizes increase selection pressure, favouring high-fitness individuals for a focused search. Smaller sizes promote diversity by allowing less fit individuals a chance to be selected, preventing premature convergence and maintaining exploration. We thus try additional values $\mu \in \{0.01, 0.5\}$ and $\theta \in \{2, 12\}$.

Through these experiments, we aim to understand the behaviour and convergence characteristics of the fish population, evaluate exploration versus exploitation, and explore the trade-off between diversity and selection pressure. This knowledge will enhance our understanding of optimization dynamics in the simulation and provide insights for future algorithmic improvements.

4 Results of Experiments

This section describes the results of all of our experiments, including evidence in the form of screenshots. We comment on the most significant strategies evolved by each of the evolutions we ran for each setting. The presented screenshots capture step 900 of the (re-run) simulation. We start by describing the baseline, which is shared among all the following experiments. Experiments follow the design described in Section 3. Note that we are not interested in concrete values of the evolved parameters. We focus on the qualitative relation between the parameters and the subsequent emergent patterns in fish school behaviour.

4.1 Baseline

We first explore the results of the baseline evolution. We observed that the resulting fish schools evolved in several different ways. The best overall parametrizations found by evolution for each of the three settings are displayed in Table 3.

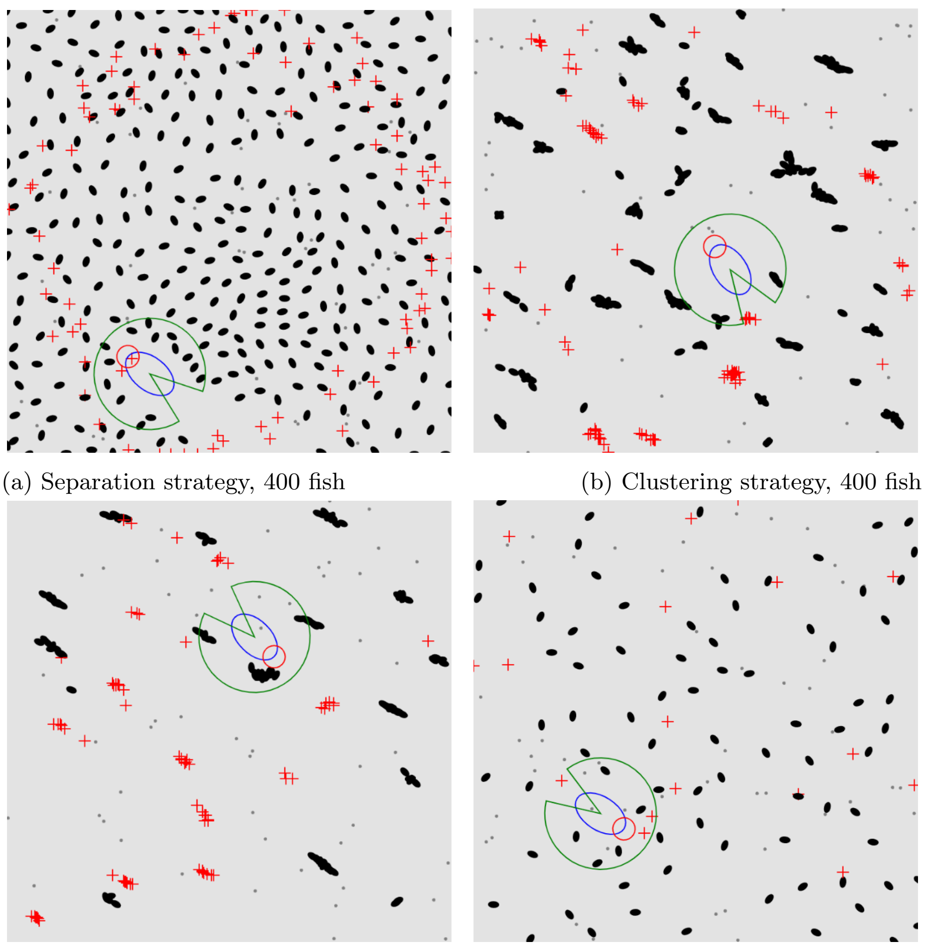
N. of fish	N. eaten	Momentum	Alignment	Cohesion	Separation	Shark rep.	Food attr.
400	66	0.644	0.179	0.016	14.286	48.094	0.138
200	60	0.300	0.196	0.592	2.821	13.565	0.928
100	13	0.300	0.428	0.038	71.542	8.367	0.632

Table 3: Overview of best parametrizations found by baseline evolution for each of the three settings.

Setting with 400 fish Two main strategies evolved during our five evolution runs. The most effective strategy (used twice) relied on very high *separation* and *shark repulsion* while keeping *cohesion* very low. This results in fish occupying the whole area but moving in similar directions as neighbours and reacting to the shark together (see Figure 2a). Distances in between give them enough time to react. We call this a **separation** strategy. Another three evolution runs resulted in a setting that did not invest into *separation* but instead used slightly higher values for other forces. It produces a **clustering** behaviour, as depicted in Figure 2b. This setting's performance was significantly worse, having almost two times more fish eaten.

Setting with 200 fish All five best-performing schools' behaviours were more or less qualitatively similar in their strategies and performance. Fish primarily relied on high *shark repulsion*, moderately high *separation* while also keeping *cohesion* values moderate (one swarm even employed high *cohesion*). *Momentum* and *alignment* were usually low. There were no extreme values. The small ratios between *separation* and *cohesion* resulted in clustering behaviour. However, in combination with high *shark repulsion*, it results in **mobile clusters** that are able to evade the shark sometimes (see how a cluster evades the shark at Figure 2c).

Setting with 100 fish Similarly to the setting with 400 fish, the two best-performing schools employed (a variation of) the *separation strategy*. This way, they used the space to disperse, mostly avoiding the shark and exploiting its momentum that does not allow it to turn fast. They did not interact together as much as in the 400 fish setting. This behaviour is captured in Figure 2d. The other prominent strategy again utilized *shark repulsion* and *cohesion* to employ *mobile clusters*, as described in the previous paragraph. However, on average, this resulted in twice as many dead fish as the separation strategy. All five schools used relatively low *momentum*, enabling bursts of movement to avoid sharks.



(a) Separation strategy, 400 fish

(b) Clustering strategy, 400 fish

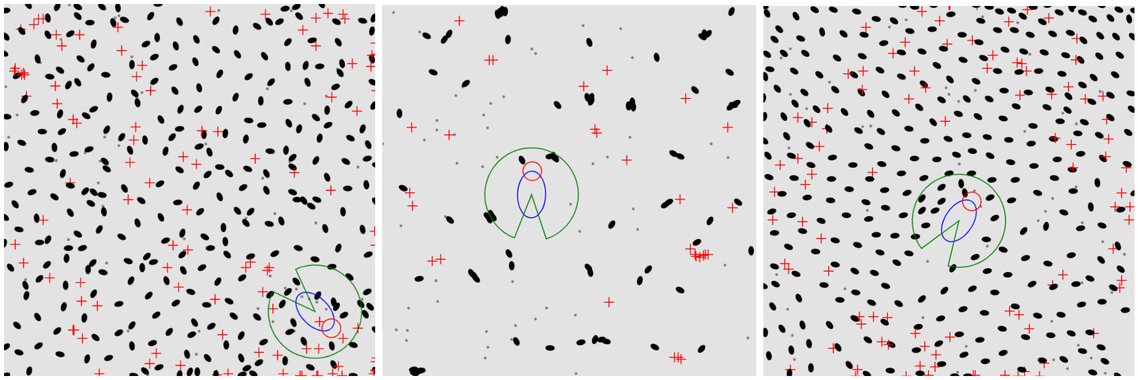
(c) Clustering strategy, 200 fish

(d) Separation strategy, 100 fish

Figure 2: Screenshots of most prominent strategies evolved by baseline evolution.

4.2 Extending Objective Function With Food

No drastic qualitative changes happened with the inclusion of food in the objective function with a weight of 0.0001. For 400 fish, successful schools employed *clustering* or *separation* strategies (separation being twice as proficient again). The separation strategy was also more successful



(a) Chaotic sep. strategy, 400 fish (b) Stimuli-based strategy, 100 fish (c) Aligned strategy, 400 fish

Figure 3: Screenshots of most prominent strategies evolved by modified evolutions (food-influenced or with different parameters) not encountered for baseline.

in food consumption even without investing into *food attraction* force simply because the fish occupied a larger space and actively moved around. For 200 fish, different variants of clustering behaviour were encountered, similar to the baseline. The 100 fish schools also behaved similarly to the baseline.

With a food weight of 0.001, the food search behaviour started to emerge in some schools and has slightly affected some of the evolutions. Including food in the objective function benefits the actively moving swarms, even if they do not evolve high *food attraction*. The most successful school of 100 fish employed a separation strategy, evolving very low *cohesion* and *allignment* parameters. For the first time, a separation strategy was employed in the setting with 200 fish (resulting in two times fewer deaths than the clustering strategy). One of the 400 fish schools employed a clustering strategy, having a high death count and not taking advantage of the food. The other two used a separation strategy and simultaneously invested heavily in *food attraction*, resulting in far better objective function values.

Finally, including the food with a weight of 0.01 significantly changed the evolution course. First, as expected, most swarms (all but one) this time invested into *food attraction*. No swarm employed a clustering strategy. However, one of the 100 fish schools employed an entirely new strategy. In this strategy, depicted in Figure 3b, fish use low *cohesion* to avoid larger clusters but do not invest into *separation*. Instead, fish use external stimuli for movement, utilizing high *shark repulsion* and *food attraction* to burst out when a shark or food appears. We thus call it a **stimuli-based** strategy. It results in slightly more deaths than separation but secures more food. All other schools employ a sort of separation strategy, sometimes with extreme parameter values. High values of *food attraction* result in more chaotic behaviour, shown in Figure 3a, which we call **chaotic separation**. Overall, high investment into *food attraction* results in slightly more deaths on average than baseline schools that used the same strategy. However, it ensures that evolution favours actively moving schools, which behave significantly better than clustering ones. Interestingly, the settings achieving the best overall values for the objective function also achieved the highest survivability.

4.3 Robustness assessment

In this section, we present the overall results of our robustness experiments. We first explore the results of conducted shark parameters experiments. The experiments regarding evolution's parameters follow. We ran the evolution procedure three times for each setting. We present the similarities or differences with respect to the baseline. Tables with detailed results are enclosed in Section D of the Appendix. Note that these experiments use evolution with a baseline objective function that does not account for food.

4.3.1 Shark Parameters

Number of sharks For schools with 400 fish, we have observed similar strategies for both settings with 2 and 3 sharks. Both had employed the *chaotic separation* (as described in 4.2) once. The other two evolved simulations utilized *clustering strategy* as in subsection 4.1. The number of eaten fish in the *chaotic separation* strategy grew linearly with the number of sharks. Like the baseline, the clustering strategy resulted in twice more eaten fish.

The results for 200 fish settings were more diverse. With three sharks, we observed two strategies - regular *separation strategy* (employed once) and *chaotic separation strategy* (employed twice). Strangely, the fish significantly relied on *food attraction*, sometimes even more than on the *shark repulsion*. The settings with two sharks twice employed the *clustering strategy* and once the *chaotic separation strategy* (the *clustering strategy* again being twice less proficient).

For the 100 fish settings, swarms again employed multiple strategies. For two sharks, each school evolved differently – one utilized *stimuli-based strategy*, one developed a *mobile clusters strategy* and one classical *clustering strategy*. Each one had a similar performance, and all performed even worse than experiments with three sharks. Three shark experiments, on the other hand, relied (more or less) on the successful *chaotic separation*.

Generally, in all experiments where evolution converged to separation strategy, the number of eaten fish per 1 shark per 100 fish was more or less the same. See Section D.2 of the Appendix for further details.

Shark's maximal speed In experiments with the shark's maximal speed, the 400 fish setting mainly employed a *clustering strategy*, relying on the fact that the shark can not effectively search for small clusters. For the simulations with the speed limit lowered to 5, one school employing *chaotic separation strategy* achieved the survival of all but three fish. Note that in some of the clustering strategies, the *cohesion* strength was larger than *shark repulsion*. Swarms that employed *clustering* suffered more than 100 deaths.

All three schools of 200 fish employed *mobile clusters strategy* when the shark's speed limit was reduced to 5. This proved successful due to the fact that the fish clusters were now able to react to a shark attack. With shark's *maximal speed* increased to 9, three different strategies evolved - *chaotic separation*, *mobile clustering* and an "aggressive" version of *stimuli-based strategy* that utilizes higher *momentum* and *food attraction* to keep moving around.

And finally, for the 100 fish setting, *mobile cluster strategy* was used twice when the shark's speed was lowered to 5. A *stimuli-based strategy*, which this time put more emphasis on *momentum* and less on *food attraction*, was able to achieve 0 eaten fish. As for the setting with shark's *maximal speed* increased to 9, we observed *clustering strategy* twice and *stimuli-based strategy* once. Each of those had similar results.

Generally, with the shark's *maximal speed* set to 9, the number of eaten fish was two times higher than compared to *max speed* of 5 since the speed is almost twice as high.

4.3.2 Selection Pressure

The selection pressure and diversity of the evolution are experimented on similarly by repeating each setting three times. The results from the experiments are summarized in the paragraphs below.

Mutation probability We experimented with mutation rates higher (0.5) and lower (0.01) than the baseline parameters. For each fish setting of 400, 200, and 100, the higher mutation rate, on average, provided better results as the evolution was able to explore more diverse settings, thus escaping local optima.
For 400 fishes, similar strategies were observed irrespective of the change in mutation rates. The fish opted to cluster up. Even though both led to almost three times more fish being eaten than baseline, more priority to food was provided with a lower mutation rate.
For 200 fish, the same observations as for the previous setting were made, except the strategy undertaken is the chaotic separation strategy. During one of the experiments, a unique strategy was found where fish mostly moved in an almost streamlined fashion parallel to one another, only changing direction when encountering the shark, as shown in Figure 3c. We call this **aligned** strategy. The parameter values are similar to the regular separation-based strategy, with higher *alignment*.
As for the 100 fish setting, results similar to the 400 fish setting were encountered with fishes clustering up together. However, no such overextension for food was encountered in either 3 of the

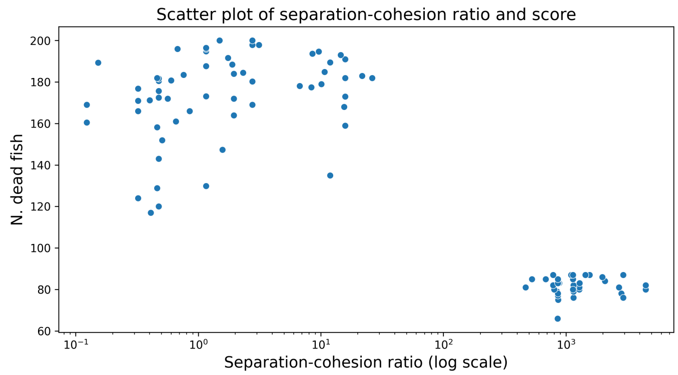


Figure 4: Scatterplot showing the relation between separation-cohesion ratio (logarithmic) and the number of fish deaths. All 400-fish schools from the last generations of each of the five baseline evolutions are included (a hundred schools in total). Clustering strategies (with a low separation-cohesion ratio, top left) have a significantly higher death toll than separation-based ones (bottom right).

experiments for the setting.

Tournament size The experimentation is performed similarly to the mutation rate with a value higher (12) and lower (2) than the baseline.
For the 400 fish setting, we could see the diversity in action, as one of the three experiments for lower tournament selection led to the chaotic separation strategy while the rest followed the clustering strategy.
As for the setting with 200 fish, the clustering strategy was followed by all three simulations with no unique differences from the baseline.
Though for 100 fish, we can see that with larger tournament sizes, the best individuals are picked, which eventually leads to low counts of dead fish compared to the lower values. Also, most of the experiments for tournament sizes evolved into clustering strategies except just one in the setting with a tournament size of 12.

5 Discussion

During the course of our experiments, several prominent modes of behaviour emerged from our evolutions. We summarize the usual parameter values of the most prominent strategies in Table 4. The two most employed types of strategies were either of separation-based nature or (some form of) clustering. The clustering behaviour emerged from different parameter settings but shared a common characteristic – low separation-to-cohesion ratio. On the other hand, the separation-based strategy had this ratio very high, often having extreme values for these parameters. Most of the settings employed low to moderate values for alignment and (as expected) high shark repulsion. More chaotic strategies employed high food attraction, perhaps even using food as stimuli for random movement. Momentum varied in most of the settings.

The fish using separation-based strategies significantly outperformed the fish that used clustering-based ones, both in terms of survival abilities and food consumption. See the relation between the separation-cohesion ratio and the number of fish deaths in Figure 4. The clustering-based strategies may, on average, "be good enough" (often, only a few clusters get eaten during a simulation) but have significantly worse worst-case scenarios (most of the clusters are eaten sometimes) and therefore are not optimal.

The two main strategies (separation and clustering) proved robust as they emerged for various settings of simulation/evolution. By including food in the evolution's objective function, actively moving swarms benefited. This results in evolution being more prone to converge to some version of the *separation* strategy. In settings with more or faster sharks, behaviour similar to the baseline was observed. However, chaotic or mobile versions of the general strategies emerged more often, helping fish escape the sharks' more or less regular movements. Experiments with various evolution

Strategy	Momentum	Alignment	Cohesion	Separation	Shark rep.	Food attr.
separation	-	low	very low	(very) high	(very) high	-
chaotic separation	-	-	very low	very high	very high	(very) high
clustering	-	low/mod	mod	low/mod	mod	-
mobile clusters	-	low/mod	mod	low/mod	(very) high	-
stimuli-based	-	very low	very low	low	very high	(very) high
aligned	mod	mod	very low	high	high	low

Table 4: Assessment of parameter values regarding encountered strategies. The scale is relative with respect to the range of values encountered for each parameter: very low, low, moderate (mod), high, or very high; the "-" sign means that the parameter seems unimportant based on experiments. Note that different combinations may result in the same qualitative behaviour; these are the most frequent settings we encountered.

settings proved valuable, as a new, *aligned* strategy was discovered, where fish cooperate on a school level and behave more "naturally". Otherwise, the explored patterns remain similar, proving their robustness.

Many aspects of our simulation (modelling) could be modified to be more realistic. The most obvious one is the shark's behaviour. Our simple hunting strategy seems sufficient for initial experiments and validating the ideas, but it could be improved in several ways. First, as the shark does not have any real memory, it may closely miss a cluster of fish (due to its inability to turn fast) and forget about it, which benefits clustering techniques. Second, the shark seems to struggle in settings where fish occupy the whole area – sometimes just hunting in circles or lines. Lastly, unlike in nature, our shark cannot produce burst movement to catch fish. Investigating and improving these aspects might prove beneficial. Similarly, fish modelling could be improved by, for instance, implementing a steering limit. Lastly, different food distribution strategies should be further studied, as food attraction proved critical for some settings.

We believe that the observed fish strategies would transfer to 3D settings as well. However, with the additional dimension, new interesting patterns may emerge, perhaps more complex and realistic. Similarly, several versions of current modes of behaviour may appear. To fully understand the difference between 2D and 3D settings, a new study would be needed.

6 Conclusion

Using an approach combining Swarm Intelligence and Evolutionary Algorithms, we studied the behaviour of fish schools in settings with predators and food. We performed a set of experiments to explore what strategies would emerge via artificial evolution. The evolution procedure helped us identify several modes of behaviour. Some produced behaviours were unrealistic, such as the ones based on clustering or involving too much chaos. These often corresponded to local minima. However, the evolution also identified several prominent strategies, such as the base *separation* or *aligned* strategies, where fish successfully cooperated on a schooling level to avoid predators.

Based on our results, we believe that combining the simulation with evolution-based optimization is a promising way to explore fish school behaviour. The evolution proved to be instrumental in identifying a diverse range of behaviours that might otherwise be missed. We believe the method is worth pursuing further, and this work can serve as a foundation for further improvements.

Even though we managed to explore several prominent behavioural patterns, there might be other successful strategies due to the vast parameter space and inherent randomness of both simulation and evolution. For this reason, further studies should involve more thorough testing. We think that exploring values of more parameters and their combinations while repeating experiments more times could possibly identify better simulation and evolution settings and potentially new strategies.

References

[1] C. J. A. B. Filho, F. B. de Lima Neto, A. J. C. C. Lins, A. I. S. Nascimento, and M. P. Lima, "Fish school search," in *Nature-Inspired Algorithms for Optimisation* (R. Chiong, ed.), pp. 261–277, Springer, 2009.

[2] Y. Luo, J. Zhang, and X. Li, "The optimization of pid controller parameters based on artificial fish swarm algorithm," in *2007 IEEE International Conference on Automation and Logistics*, pp. 1058–1062, 2007.

[3] J. Hu, X. Zeng, and J. Xiao, "Artificial fish school algorithm for function optimization," in *2010 2nd International Conference on Information Engineering and Computer Science*, pp. 1–4, 2010.

[4] C. C. Ioannou, "Swarm intelligence in fish? the difficulty in demonstrating distributed and self-organised collective intelligence in (some) animal groups," *Behavioural Processes*, vol. 141, pp. 141–151, 2017. The Cognition of Fish.

[5] J. M. Gardiner, J. Atema, R. E. Hueter, and P. J. Motta, "Multisensory integration and behavioral plasticity in sharks from different ecological niches," *PLOS ONE*, vol. 9, pp. 1–13, 04 2014.

[6] M. McComb-Kobza, *Visual Adaptations in Sharks, Skates and Rays.* PhD thesis, Florida Atlantic University, 08 2009.

[7] A. Silva, A. Neves, and E. Costa, "An empirical comparison of particle swarm and predator prey optimisation," in *Artificial Intelligence and Cognitive Science*, pp. 103–110, Springer, 2002.

[8] L. Dioşan and M. Oltean, "Evolving the structure of the particle swarm optimization algorithms," in *Evolutionary Computation in Combinatorial Optimization*, pp. 25–36, Springer, 2006.

[9] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, p. 25–34, Association for Computing Machinery, 1987.

[10] R. Azizi, "Empirical study of artificial fish swarm algorithm," *International Journal of Computing, Communications and Networking*, vol. 3, no. 1, 2014.

[11] J. Teghem, "Metaheuristics. from design to implementation," *European Journal of Operational Research*, vol. 205, no. 2, pp. 486–487, 2010.

A Code and Results Availability

The implementation is available at `https://github.com/ondrej33/Natural-computing`. The repository contains the source code for the algorithms used, the visualization tool, as well as results. The main part of the implementation consists of highly optimized C++ simulation, parallelized evolution in Python, and JS visualization utilizing a complex logging system. Further, we include several animations depicting various strategies. All the instructions and repository documentation are given in a README file.

B Author Contributions

Ondřej Huvar:

- Did a large part of the project design and research
- Designed and implemented half of the simulation
- Designed and implemented the evolution procedure
- Designed methodology for experiments, conducted baseline experiments and experiments regarding the objective function
- Participated in making flash talk slides and presenting

- Fully wrote the following parts of the report: related work (1.1), evolution (2.2), experiments with baseline (4.1) and with the objective function (3.1, 4.2), discussion (5) and conclusion (6)
- Partially wrote other parts of the report – introduction, the general design of experiments, and helped with rewriting the parts on simulation and results

Tomáš Kalabis:

- Contributed to the research
- Designed and implemented half of the simulation
- Designed and implemented visualization
- Designed and conducted experiments regarding sharks
- Participated in making flash talk slides and presenting
- Wrote the parts in the report related to the experiments with sharks and the methodology of the simulation, proof-read the rest

Ariyan Tufchi:

- Contributed to the research
- Experimented with various evolution algorithms(universal selection) and parameters
- Designed and conducted experiments regarding evolution's parameters
- Wrote the part of the report related to the experiments with evolution's parameters and introduction.
- Researched fish and shark parameters for the basic visualization model.

C Algorithm Pseudocodes

For completeness, we present two high-level pseudocodes for algorithms involved in our evolution. The pseudocode for the whole evolution process is given as Algorithm 1. The high-level pseudocode for the reproduction strategy is given as Algorithm 2.

Algorithm 1: High-level pseudocode of the evolutionary algorithm we use.

```
Function EVOLUTION(G_max)
    P ← generate_initial_population();
    G ← 0;
    repeat
        G ← G + 1;
        evaluate(P);
        P′ ← reproduction(P);
        evaluate(P′);
        P ← replacement(P, P′);
    until G = G_max;
    return fittest_indiv(P)
```

D Further experimental data

In this section, we provide the extended version of the results of our experiments. The tables contain parameters and scores of a top-performing fish school for each setting we experimented with.

Algorithm 2: High-level pseudocode of the reproduction strategy we use.

```
Function REPRODUCTION(P)
    offsprings ← crossover(P);
    P ← concat(P, offsprings);
    P ← generate_n_copies(P);
    P ← mutate(P);
    P ← filter_redundant(P);
    return P;
```

D.1 Experiments with objective function

We provide the best parameterizations regarding our experiments with objective function in Table 5.

N. of fish	Food weight	N. eaten	Food eaten	moment.	alignment	cohesion	separation	shark repulsion	food attr.
400	*0.01*	*70*	*37516*	0.351	0.457	0.019	100.116	42.187	9.461
200	*0.01*	*30*	*18588*	0.342	0.122	0.012	53.191	8.075	5.493
100	*0.01*	*20*	*899*	0.449	0.046	0.256	0.631	8.595	6.979
400	*0.001*	*75*	*35718*	0.388	0.049	0.049	82.294	45.582	5.0877
200	*0.001*	*28*	*8692*	0.413	0.148	0.005	10.489	2.004	1.210
100	*0.001*	*12*	*3061*	0.549	0.092	0.006	9.136	7.347	0.820
400	*0.0001*	*67*	*12451*	0.341	0.063	0.114	60.163	18.462	1.123
200	*0.0001*	*48*	*148*	1.000	0.431	0.588	0.696	11.90	0.683
100	*0.0001*	*21*	*829*	0.400	0.094	0.262	1.228	12.825	4.478

Table 5: Overview of the best parametrizations found by evolution for each of the three settings of objective function involving food.

D.2 Experiments with number of sharks

We provide the best parameterizations regarding our experiments with the number of sharks in Table 7. Also, in Table 6 is captured number of eaten fish per 100 fish per 1 shark for each conducted experiment.

Number of fish	400	400	400	200	200	200	100	100	100
Number of sharks	1	2	3	1	2	3	1	2	3
N. eaten	66	123	175	60	69	104	13	56	41
N. eaten per shark per 100 fish	16	15	14	30	17	17	13	28	13

Table 6: Table captures the number of eaten fish per shark per 100 fish in each conducted experiment with a number of sharks.

D.3 Experiments with a shark's maximal speed

We provide the best parameterizations regarding our experiments with the max speed of a shark in Table 8.

D.4 Experiments with mutation probabilities

Best parameterizations regarding our experiments with the mutation rate are presented in Table 9.

N. of fish	N. of sharks	N. eaten	momentum	alignment	cohesion	separation	shark rep.	food attr.
400	*1*	*66*	*0.644*	*0.179*	*0.016*	*14.286*	*48.094*	*0.138*
400	2	123	0.300	0.335	0.019	25.010	23.746	1.686
400	3	175	0.300	0.126	0.016	12.259	15.367	0.951
200	*1*	*60*	*0.300*	*0.196*	*0.592*	*2.821*	*13.565*	*0.928*
200	2	69	0.454	0.079	0.044	26.068	9.361	1.919
200	3	104	0.300	0.129	0.020	11.136	18.984	10.095
100	*1*	*13*	*0.300*	*0.428*	*0.038*	*71.542*	*8.367*	*0.632*
100	2	56	0.360	0.252	0.682	3.938	16.308	0.904
100	3	41	0.300	0.041	0.024	28.012	7.594	1.862

Table 7: Overview of the best parametrizations found by evolution for each of the nine settings with varying *Number of sharks* and *Number of fish*.

N. of fish	Max speed	N. eaten	momentum	alignment	cohesion	separation	shark rep.	food attr.
100	5	0	0.588	0.019	0.030	16.317	7.845	0.292
	7	13	0.300	0.429	0.039	71.542	8.368	0.633
	9	38	0.309	0.374	0.401	0.736	10.009	0.102
200	5	32	0.498	0.233	0.385	2.386	19.596	0.036
	7	60	0.300	0.196	0.593	2.822	13.566	0.929
	9	62	0.300	0.0382	0.022	10.946	14.323	2.069
400	5	3	0.757	0.009	0.053	127.397	10.968	1.231
	7	66	0.644	0.179	0.017	14.286	48.094	0.139
	9	167	0.387	0.142	9.971	0.264	0.989	11.775

Table 8: Overview of the best parametrizations found by evolution for each of the nine settings with varying *Shark max speed* and *Number of fish*.

N. of fish	Mutation Probab.	N. eaten	momentum	alignment	cohesion	separation	shark rep.	food attr.
400	*0.5*	*215*	0.88853	0.62086	0.53153	1.88214	20.82389	0.18616
200	*0.5*	*25*	0.31269	0.03834	0.00162	26.24291	19.57387	0.06204
100	*0.5*	*32*	0.31309	0.04143	0.21609	6.47320	12.37337	0.16524
400	*0.01*	*232*	0.30000	0.85059	0.02432	0.28768	8.72376	5.67725
200	*0.01*	*49*	0.42880	0.90255	0.01992	25.93555	9.08738	0.31899
100	*0.01*	*42*	0.38982	0.07853	0.28327	0.47898	6.61570	0.13606

Table 9: Overview of the best parametrizations found by evolution for each of the three settings with varying *mutation rates* and *Number of fish*.

D.5 Experiments with tournament sizes

Best parameterizations regarding our experiments with the tournament size are presented in Table 10.

N. of fish	Tournam. size	N. eaten	momentum	alignment	cohesion	separation	shark rep.	food attr.
400	*12*	*274*	0.87268	0.18021	0.29487	0.14791	9.42974	0.00862
200	*12*	*101*	0.37845	0.59621	0.98388	0.61950	13.10340	0.58827
100	*12*	*25*	0.30000	0.27354	0.00310	0.30728	9.86888	1.45142
400	*2*	*77*	0.38087	0.18416	0.00870	13.62531	11.67065	0.39792
200	*2*	*40*	0.37211	0.005511	0.04183	6.64446	15.15962	4.92450
100	*2*	*71*	0.44710	0.01659	0.57917	0.83480	25.19391	0.24794

Table 10: Overview of the best parametrizations found by evolution for each of the three settings with varying *tournament sizes* and *Number of fish*.