

Recovering Hidden Swarm Parameters Using a Simulated “Robofish”

Richard Coates¹ and Simon Hickinbotham¹

¹YCCSA, Department of Computer Science, University of York, Heslington, York YO10 5GH, UK
rdc501@gmail.com sjh@cs.york.ac.uk

Abstract

Swarming is behaviour which emerges from the action of individual agents. Models of swarm behaviour impose fixed model parameters on the agents comprising the swarm. This paper evaluates the possibility of extracting the parameters of a swarm model from the swarm. This can be achieved by evolving the parameters of a single agent that interacts with the swarm. The approach was inspired by work on so-called “robofish” by Faria et al. If we assume that the collective dynamics of wild animals can be modelled, it would be desirable to recover the dynamics of the model via interaction with them. We demonstrate that it is possible to recover the parameters of a shoaling model used by a swarm. We present an evaluation of this approach, using a genetic algorithm to drive the learning process. The experiments also reveal information about the effects of varying the parameters of the model on the emergent swarm dynamics.

Introduction

In nature many animals travel in flocks, shoals, swarms and other large groups. Several models have been proposed to replicate this phenomena (Aoki, 1982; Reynolds, 1987; Couzin et al., 2002). In each of these models, individuals follow local rules which produce the swarm as an emergent phenomenon. However little work has been done to find out the validity of these models and how accurately they map to behaviour in real-world swarms. In this paper we explore this possibility by investigating whether an agent can recover the parameters of a swarm by monitoring its interactions with the rest of the swarm. We test an evolutionary approach to this learning problem. If the evolved behaviour of the agent and swarm is identical then the models used by them are functionally equivalent. This novel approach was inspired by the work of Faria et al. (2010), in which a robotic fish (or *robofish*) interacts with a shoal of sticklebacks *Gasterosteus aculeatus* L. We refer to the shoal of fish with which the robofish interacts as the *modelfish* in the rest of this paper, since it is presumed that the shoal is following a model of behaviour which the researcher is trying to discover. This paper extends the work presented in (Coates and Hickinbotham, 2011).

It is only possible to determine whether model parameters can be learned in this way if the parameters of the swarm are known. Accordingly, we test the approach in simulation, where a swarm of modelfish follow a pre-specified model. In addition, the fitness function which is used to evolve the robofish model only makes use of observations about the emergent behaviour of the swarm. In other words, the model parameters are “hidden” from the evolutionary algorithm.

Aoki (1982) described one of the first attempts to accurately model the behaviour of fish. The behaviour of each fish is determined by the position of its neighbors. The space around the fish is divided into zones of perception relative to the position and orientation of the fish. Each zone has a corresponding behaviour linked to it, commonly called a *compulsion*. The presence of fish in a particular zone increases the contribution of the compulsion to the behaviour of the fish in the next discrete time step. In Aoki’s model, the fish’s area of perception is split into three radii. Behind the fish there is also a “blind spot”, the contents of which do not contribute to the fish’s behaviour. Nearest the fish is the zone of repulsion, then comes the zone of orientation and finally the zone of attraction. Aoki’s model only takes into account up to four neighboring fish in the zone of perception. These are selected randomly (with a greater chance of choosing those in front) and have a diminishing effect of the movement of the fish. If no neighbours are found in the zone of perception, the fish will try and move towards any it can see, no matter the distance. The neighbours it finds (and the zones they appear in) change the mean and standard variation of a Gaussian model which governs the update of the heading of the fish.

Swarm simulations were originally developed to study the behaviour of real animals. They were used to provide insights into possible reasons for why swarms behave as they do and how changing the simulations parameters affects the resulting behaviour. Since then they have found uses in computer graphics (Reynolds, 1987), as well as in searching data (Kennedy and Eberhart, 1995). Reynolds (1987) developed one of the best known swarm models, designed not necessarily to be authentic but to give aesthetically pleasing move-

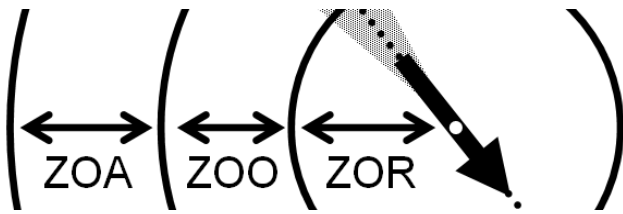


Figure 1: The three zones of the 2D Couzin model. The orientation of the fish is shown by a bold arrow (centre). The blind spot is shown as a shaded area. The widths of the zones of Attraction (ZOA), Orientation (ZOO) and Repulsion (ZOR) are indicated by arrows. Note that the position of the outer zones relative to the fish are dependent upon the widths of the inner zones.

ment for animation purposes. It consists of a single zone of perception within which a “boi” is able to detect others. It’s movement is also governed by three compulsions with regards to the neighbours it can see, the urge to move towards a neighbour, away from them or in the same direction as them. These compulsions are combined with the position and heading of the neighbours to provide a new heading for the boi.

The Couzin model shares much with Aoki’s model. It retains the three zones and their effect on the heading of the fish. However like Reynolds’ model fish outside of the range of perception have no effect at all. In addition, in a single time step, only one of the zones ever has an influence on the behaviour of the fish. The zone of repulsion (ZOR) takes precedence over the other two compulsions. If a neighbour is found in this zone then the other two are ignored and the fish will just try and swim away from neighbours in the ZOR. If no fish are found in the ZOR then only the fish in the zone of orientation (ZOO) are considered in the update of the heading. Modelfish in the zone of attraction (ZOA) are only considered if the ZOR and ZOO are empty. We note here the similarity of this configuration with Brooks’ subsumption architecture (Brooks, 1999).

The above swarming systems seem very different but they have the same foundations: they are built on simple rules, and the complex behaviour they exhibit is a by product of these, not explicitly stated by them; they involve extensive interaction between individual members of the swarm using observed information about the other individuals; and the behaviour is highly dependent on the parameters used.

Genetic algorithms (GAs) use emulations of evolution in order to solve computational problems which may be difficult to solve using more conventional techniques Goldberg (1989). The main components are a genome which specifies how the individual is represented in the algorithm, a phenotype which is how the genotype maps onto real world attributes, a fitness function which states how good each individual is and functions to perform mutation and crossover

to produce offspring. Crossover focuses the search on good values by taking two parents and recombining their genomes to create a child; mutation adds a random element to the children which increases variety in the population.

Some work has been done to use genetic algorithms to modify the behaviour of swarms. Conley (2005) used a genetic algorithm to tune a particle swarm optimisation (PSO) search. Geoboids (Macgill and Openshaw, 1998) was the basis of this work, where the swarm moves over a landscape looking for clusters of points, returning the locations of all the clusters it thinks it has found. Conley used a hierarchical fitness function and tournament selection to avoid having to assign each individual an absolute fitness value. Instead, two individuals are compared using a series of criteria, if one wins out on a test then the function stops otherwise the next test is carried out. The tests compare: the number of clusters found; the ratio of distinct clusters to total clusters; the number of dead boi”s (those which are in empty regions of the dataset when the algorithm finishes); the number of comparisons made (adjusted for the size of the flock) and finally the amount of the dataset searched.

Kwong and Jacob (2003) used genetic algorithms to change many parameters used by a swarm in order to induce “desirable behaviour”, that is swarms which moved in a certain way. They had no automatic fitness function to guide the GA and instead assigned a fitness to each swarm created by the GA by eye, based upon observing its behaviour as it ran. This type of fitness measure could easily be adapted for our work, but it is prone to human error when the differences between individuals are subtle (although some research has been done on methods to improve this (Khemka et al., 2009)).

(Stoops et al., 2010) examined the rules that swarms adhere to by using data mining and rule classification algorithms. A basic experiment into this issue involved running a swarm simulation based on Reynolds’ boi”s and recording data related to the individual’s movement (the time, position, heading, speed and separation and detection radii). A rule classifier was run on this data to create a series of potential rules. These were then tested by running the boi”s again but with these rules instead of the defaults. This run was compared to the original and used to modify the rules.

Methodology

The experiments we report here used a two-dimensional version of the Couzin model (Couzin et al., 2002; Wood and Ackland, 2007) of shoaling behaviour as the basis of simulated fish movement. Like many of the other proposed models it establishes zones of perception around each fish (see figure 1). The location of neighbours within these zones governs the fish’s future movement (being attracted to neighbours in the ZOA, oriented with neighbours in the ZOO, and repelled by those in the ZOR). Our experiments were designed to determine whether it was possible to learn the pa-

parameters of a swarm of modelfish running a known model by adding an agent (robofish) to it and monitoring its behaviour relative to the modelfish. As the robofish behaviour is specified via the same underlying model as the swarm, it follows that if the correct parameters for the robofish model can be set appropriately then its behaviour should be indistinguishable from that of the modelfish. The parameters used by each robofish were encoded in the chromosome of a genetic algorithm. The GA requires a fitness function to evaluate the robofish after a run, which will summarise the robofish's interaction with the modelfish into a single numerical value. The aim of the fitness function is to accurately represent how similar the behaviour of the robofish is to the modelfish. We hypothesised that this should allow the GA to converge on the parameters used by the modelfish, under the central assumption in this work that similar parameters alone will allow a robofish to exhibit behaviour identical to a modelfish.

The motivation for the design of our fitness function was to determine if the robofish was interacting with the modelfish or travelling the arena independently. Accordingly, the experiments used the average Euclidean distance from the robofish to each of the modelfish as the fitness function. This strategy is based on two assumptions: (1), that similar, but not identical behaviour will allow the robofish to interact with the swarm (smooth fitness landscape); (2) that only one configuration of the swarm model will induce this behaviour (no local optima). It is clear that the fitness function needs to be appropriate to the learning task at hand – the parameters of the model are assumed to be impossible to estimate directly, so we can only use the emergent behaviour of the modelfish swarm as the basis for our fitness function. Since this is such an important issue, we took steps to evaluate the fitness measure with respect to the model parameters in the GA.

Experiments

A single robofish was used in each simulation to ensure that the robofish's behaviour was determined only by its internal model and interaction with the modelfish. If more were used there would be potential for the robofish to interact each other and produce behaviour that was not programmed into the modelfish. In these experiments the only parameters to be changed are the sizes of the three zone widths ZOA, ZOR and ZOO. These were chosen as they control the main aspects of the fish's behaviour.

The parameters used by the model fish in the swarm was set throughout the experiments with parameters used in Wood's original experiment (Wood and Ackland, 2007) (see table 1). This seemed to give an aesthetically realistic swarm whose behaviour often transitioned between different types (Kwong and Jacob, 2003). This would prevent the evolution of robofish who could only swim like the others in certain swarm configurations. With the robofish sharing the same model as the modelfish, the chance of this happening

Parameter	Value
ZOR	1
ZOO	12
ZOA	13
Velocity	2.25
Blind spot	90°
World size	240
Number of modelfish	99
Number of robofish	1
Warmup time	5000
Sample time	5000
Samples	100
Mutation rate	5%
Crossover rate	90%

Table 1: Default values used throughout the experiments

is small. Not much can be done about this before the experiments are run as the behaviour created by adding a robofish to the swarm was unknown *a priori*.

A problem with using these parameters is the repulsion radius ZOR=1. Since setting it to 0 effectively removes this type of perception for the robofish, it is only possible to test what happens when the robofish's ZOR is either larger than the modelfish's or absent (the radius is an integer value). This does place a limitation on the conclusions that can be drawn from the experiments but to change these values would require additional experimentation to find other configurations which produce similar behaviour in the swarm.

The arena dimensions were also the same size as in Wood's experiment: a square 240x240. This means that each modelfish can perceive around 3.7% of the world. For all the experiments in this section the number of fish in each trial (inclusive of the robofish) was set to 100. Preliminary experiments indicated that smaller numbers of fish had a tendency to form multiple groups which either never coalesce into a single swarm or do so only after a very long time interval. It also appears that it is easier for modelfish in a small swarm to escape a group altogether and swim by themselves.

To avoid any bias in the initialisation process, the swarm was run for 5000 time steps before 5000 time steps of monitored behaviour. Here the state of the system was sampled every 50 time steps. This regime reflects the dynamic nature of swarms and reduces the risk of rare, unrepresentative states distorting the fitness estimation. A robofish has to perform well for the entire run to obtain a good fitness score. A robofish which "loses" the swarm or keeps its distance at times will score poorly. In addition, each configuration was performed 5 times to further reduce the effect of the variability in runs on the analysis. This means that each robofish configuration was sampled a total of 500 times.

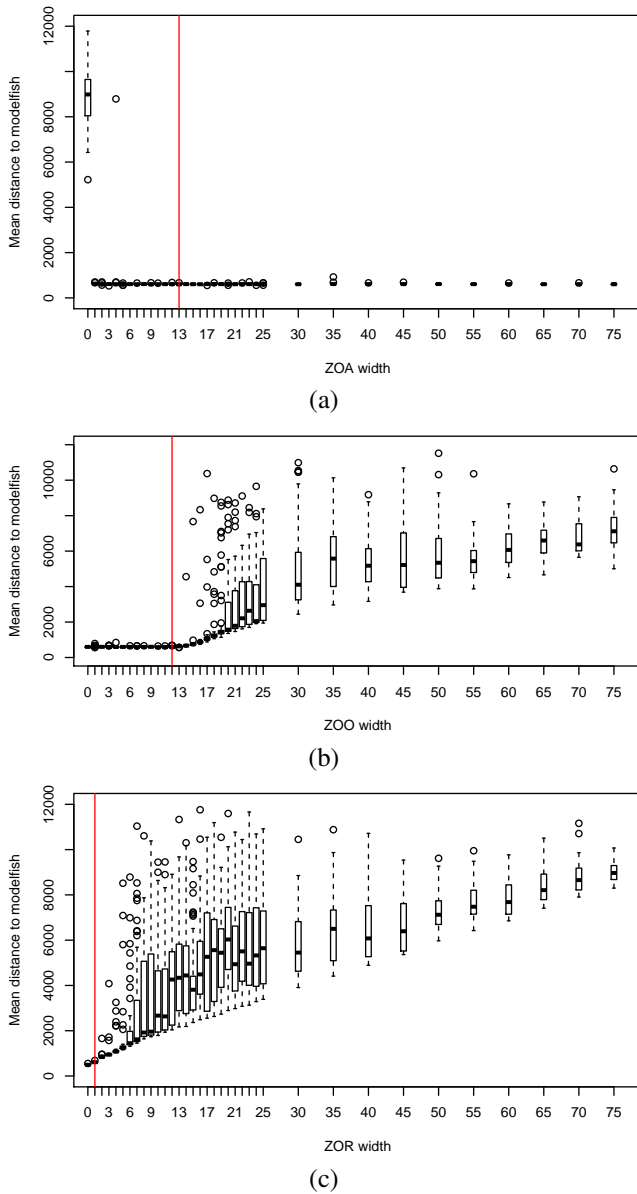


Figure 2: Effect on fitness of changes in widths of (a) Zone of Attraction, (b) Zone of Orientation, and (c) Zone of Repulsion for the robofish. The vertical red line on each plot indicates the target width value used by the modelfish.

Fitness of zone widths

Our first experiment explored the effect of changing the value of individual zone widths on the fitness of the robofish. The position of the zones relative to the fish are interdependent, as shown in figure 1. Changing one of the widths whilst keeping the other two fixed gives us a clearer understanding of the contribution of the zone width to the emergent shoal dynamics.

Each of the three zone widths were changed from zero

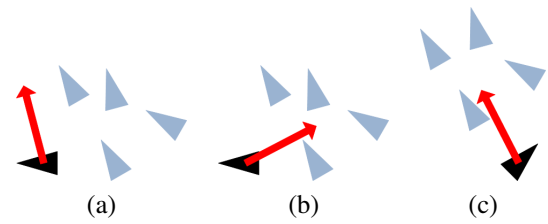


Figure 3: Robofish movement (a) at time t with a ZOO, (b) at time t without a ZOO, and (c) at time $t+1$ without a ZOO. In both cases, the robofish maintains contact with the swarm.

(eliminating the behaviour completely) to 25 (roughly double the modelfish values for ZOO and ZOA) in intervals of 1 to examine how the fitness varies around the values for the parameters actually used by the other fish. The tests were then extended using values from 30 to 75 (roughly six times the modelfish value) in intervals of 5 to explore more dissimilar configurations. Each configuration was tested over 50 trials to allow an accurate representation of the fitness of that value. Figure 2 shows the results of these experiments.

Zone of Attraction From figure 2(a) it appears that varying the ZOA has very little effect on the fitness of the robofish. However, when there is no ZOA (i.e. the width = 0), the average distance to the swarm is high, with large variance. The lack of any compulsion to swim towards other fish leads to the robofish being lost from the modelfish swarm. If the ZOA is present, no matter what the size, the robofish performs very well with a low average distance to the modelfish. It would be expected that a high ZOA would allow the robofish to find the swarm more easily if it became separated from the group, but the plot indicates that even a small ZOA with a width of 1 is effective in maintaining the robofish's contact with the swarm. This suggests that the ZOA has a role in limiting the chance of *escape* from the swarm, rather than directing behaviour to *seek* the swarm when swimming alone. Where a ZOA is present in a trial, the swarm has usually formed and contains the robofish by the time fitness measurements commence.

Fish in our model move at 2.25 units per time step. A fish on the margins of the swarm might only have contact with the swarm at the very perimeter of its zone of orientation. There is a small chance that this contact would be lost, and the fish might then escape the swarm if there was no zone of attraction at all. However a very small ZOA would be sufficient to induce the fish to move back towards the swarm should this occur, since those fish which were in the ZOO of the fish in the previous time-step will either still be in the ZOO or have moved to the ZOA – it is very unlikely that they will be completely out of range.

Note also that our fitness measure only samples the position of the robofish after a period of 5000 time steps. If we began to measure fitness from the initialisation period, we

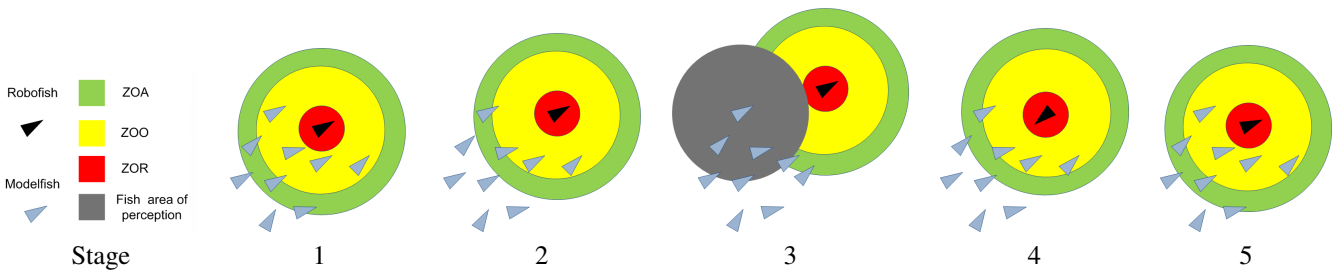


Figure 4: Effects of an individual's large ZOO on shoaling interactions. Stage 1 - The robfish is at the head of the shoal. Stage 2 - The large ZOO allows the Robofish to pull ahead. Stage 3 - The swarm is entirely in the Robofish's ZOA but the robfish cannot be seen by the swarm. Stage 4 - The robfish swims back towards the shoal. Stage 5 - The robfish has rejoined the shoal

may see the effects of varying the ZOA during the process of swarm formation.

Zone of Orientation As shown in figure 2(b), the ZOO has a much clearer effect on the fitness of the robfish. At low values for ZOO the robfish performs very well. It is possible that the interplay between ZOA and ZOO (described above) is the reason for this. The behaviour is illustrated in figure 3, where the arrow indicates the movement vector for the robfish in the next time step. In the model at time step $t + 1$ the ZOA allows the fish to move towards its neighbours, ZOO in the same direction of them. If ZOO is disabled the ZOA will still move towards where the neighbours are which is the same as moving in the direction they were facing at time t . As the time steps in the simulation are very small and the velocities of the fish small compared to their range of perception (a fish can move a maximum of 2.25 units per time step whereas its radius of perception (ZOA + ZOO + ZOR) is 26 units) the difference between moving towards the heading of the neighbours at time t and $t + 1$ is very small resulting in almost identical behaviour.

The mean distance between the robfish and the model fish increases when the robfish ZOO width rises above the ZOO width of the model fish. Between 14 and 19 a slow phase change occurs from a fairly uniform fitness to a more variable, but generally less fit behaviour afterwards. We suggest that this phase change is a by-product of the behaviour a larger ZOO induces. We hypothesise the following behaviour pattern, illustrated in figure 4. The fish in the swarm constantly change relative position in the swarm as they move, due to the stochastic element of the Couzin model. Each fish therefore spends some of the time at the front of the swarm. When the robfish is at the front of the swarm, a situation arises in which the modelfish are in the ZOO of the robfish. At the same time, the robfish is in the ZOA of some of the modelfish, but since other modelfish are in the ZOO of these modelfish, the position of the robfish is ignored by the shoal (stages 1-3 of figure 4). The robfish then changes behaviour, and swims toward the swarm (stages 4-5 of figure 4). As the width of the ZOO increases, there is

an increasing likelihood that the robfish will not successfully rejoin the swarm, since it will be further away from the swarm, and the swarm is more likely to have changed direction.

An increase in the robfish's ZOO increases the distance away from the swarm that it can travel before heading back and allows the robfish to lead again at a point further from the swarm (at lower ZOO values it will rejoin instead). Both these tendencies cause the average distance from the other fish to increase as shown on the graph in figure 2(b). However this does not directly explain the increased variation seen as the ZOO increases. The variation is due to the robfish losing the swarm. The conditions for this seem to be when the robfish has left the sight of the swarm but both the swarm and robfish are moving in almost the same direction. A small deviation from these parallel headings can place members of the swarm within the robfish's ZOA in a single time step, resulting in a situation where the robfish follows the swarm but does not influence it. As the ZOO of the robfish increases, the chance of this happening increases, and small changes in direction cause the robfish to lose sight of the swarm more often. We also postulate an increase in the ability of the robfish to find the swarm again (due to the increased area of the world the robfish can now see). Overall then, as the chances of losing the swarm increase so does the chance of the robfish finding it again, resulting in a lower jump in penalty for losing the swarm. The highest numbers tested show the range of scores decrease again whilst the mean climbs further. At this point the robfish constantly loses the swarm, resulting in uniform but unfit behaviour.

Zone of Repulsion As shown in figure 2(c), the ZOR width seems to have a much stronger and simpler effect on fitness than the other two variables. Simply put, the larger the ZOR the further away the robfish is from the others in the swarm. Although the upwards trend is visible from the start, a phase transition of the kind seen with ZOO is shown from around 3 to 7. Once again this is where the robfish starts to lose the swarm on an increasing number

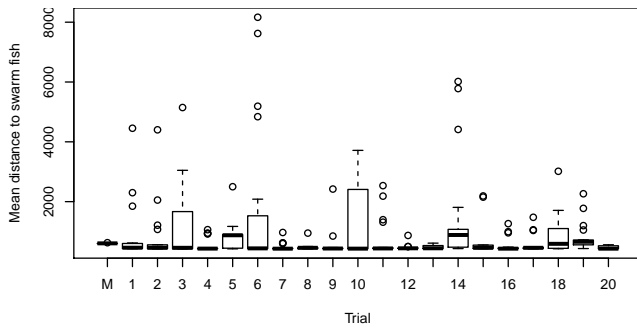


Figure 5: Distribution of fitness at end of 50 generations of 20 trials of the GA. Column 'M' indicates the distribution of fitness for the modelfish

of runs, showing increasing variance in the mean distance to the shoal fish.

Evolving the model parameters

The previous section demonstrated a relationship between changes in the individual model widths and mean distance to the shoal fish. In this section, we show how a genetic algorithm can be used to find combinations of widths for the ZOA, ZOO and ZOR that minimises the mean distance to the swarm.

We maintained a population of 20 robofish models per generation throughout the trial. This was a compromise between having a large population size (which could explore a sizable portion of the solution space) and minimising the computation time. This is important as each genome must be tested independently since we can only have one robofish evaluation per swarm, meaning that the genetic algorithm will run very slowly. Each robofish configuration was evaluated 5 times.

It was not our intention to evaluate a new configuration of a genetic algorithm. Accordingly, we implemented our genetic algorithm using PyEvolve (Butterfield et al., 2004). Crossover was set to 90%, and mutation (occurring with a 5% chance) changed the Zone widths following a Gaussian distribution with variance of 5% of the current value. The genetic algorithm was run for 50 generations. This was repeated 20 times to estimate the consistency of convergence.

The robofish's genome consisted of the widths of the three zones which were to be modified; the ZOA, ZOO and ZOR. These were stored as integers.

The three zone widths were initialised with random integers in the range [0,75]. The Pyevolve default tournament selection method was chosen to select fit individuals for subsequent generations.

Figure 5 shows the final distribution of mean distance to swarm fish for the robofish population in the final generation of the 20 runs of the genetic algorithm. The mean distance to shoal fish for robofish with the same width values as the

model fish is shown in the column marked 'M'. It can be seen that trials 3, 5, 6, 10, 14 and 18 have not fully converged, but the other fourteen trials show that the genetic algorithm has successfully reduced the mean distance to the swarm fish, as specified by the fitness function. Those trials which did not converge were composed of a mixture of individual models with a combination of low and high mean distances to the modelfish, indicating that it is likely that the runs would eventually converge to low average mean distances across the population if allowed to run for longer.

Note that the mean distance to shoal fish was higher in the control robofish 'M' that used the model zone widths. This reveals an issue with the fitness function - it was designed to evolve a fish that interacted with the shoal, but there is nothing in the fitness function to induce the evolved robofish to *mimic* the behaviour of the model fish. This is why the mean distance is reduced to a minimum, rather than converging on the value that the modelfish parameters generate.

To further illustrate the effect of the fitness function, figure 6 illustrates the change of fitness of the robofish configurations for trial 19, along with corresponding distributions of the three zone widths. In this trial, the genetic algorithm is effective at reducing the mean distance to the swarm. The target widths for the ZOA, ZOO and ZOR are shown as a red line on the bottom three figures.

The width of the ZOA shows the biggest difference between the evolved value and the model value. This is not surprising for two reasons. Firstly, as shown in figure 2(a), the ZOA has little effect on the mean distance to shoal, so it is free to vary. What might explain the drift of ZOA width to such a large value? We assert that a large ZOA is useful in allowing the robofish to find the swarm reliably. If the robofish has not found the swarm by $t = 5,000$, then its chance of being selected for the next generation will be reduced.

Both the ZOO and ZOR widths are much more similar to the model values. We claim that the value of the model ZOO is learned effectively by the GA. The ZOR tends to converge to a value of zero, compared with the model value of one. This is a small error, but the consistency of the error leads us to conclude that this value allows the robofish to have a lower mean distance to the modelfish than the modelfish have to each other.

Finally, we note that the ZOO+ZOR widths for the evolved robofish are almost identical to those for the modelfish. This implies that the outer radius of the ZOO, as specified by ZOO+ZOR is a critical parameter in determining the mean distance to other fish.

Conclusions

The work presented here has started to explore the ability of genetic algorithms to extract important parameters related to a swarm's behaviour. It is only the start of what is possible in this field. Many more aspects can be explored in relation

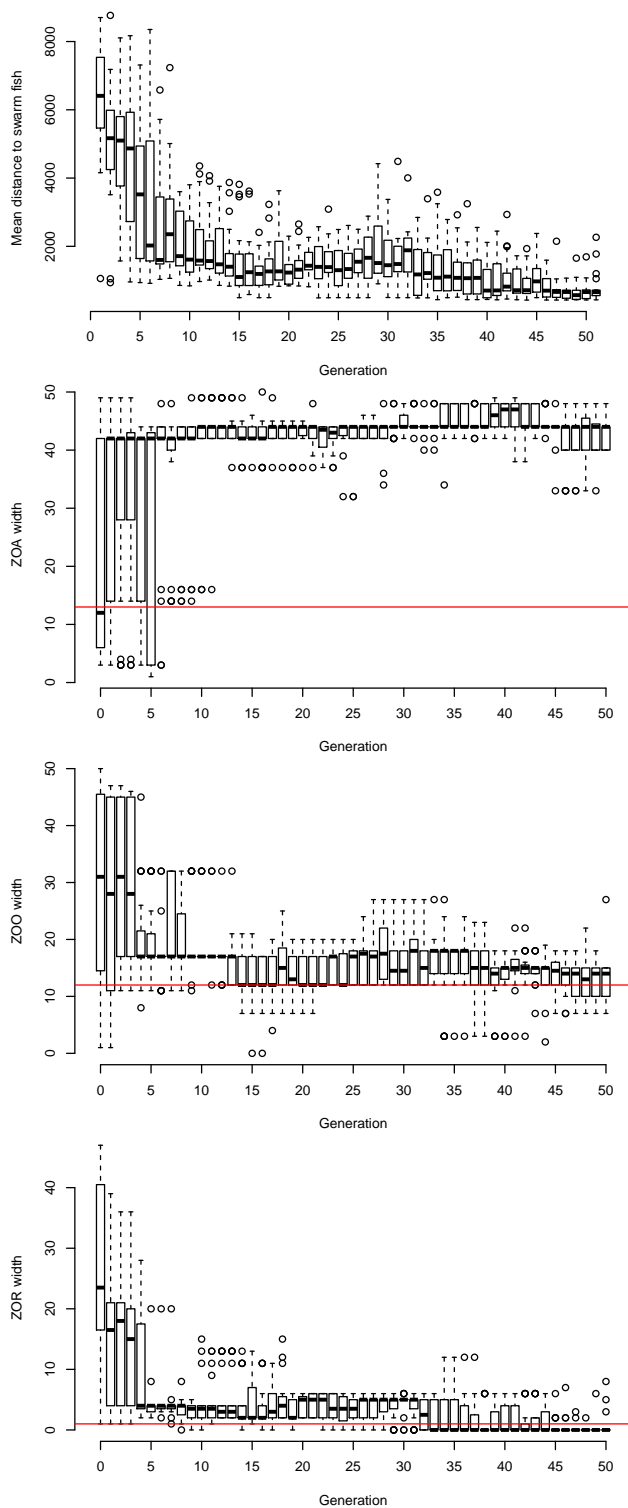


Figure 6: Run 19 of the GA. Plots from top to bottom: Fitness; Convergence to ZOA; Convergence to ZOO; Convergence to ZOR

to the goal of estimating swarm behaviour models via the interaction of configurable agents with swarm agents.

It is well known that the interactions between the zones of behaviour in the Couzin shoaling model lead to an emergent swarm. In this paper, we have investigated ways of recovering the underlying model parameters of the swarm indirectly, via the interactions of an individual with the swarm. The long-term goal of this work is to fit models to observations of shoaling of wild animals. Our initial trials, varying only one of the zone widths whilst holding the others constant, revealed the following observations:

- A ZOA is needed to produce a swarm (i.e. the ZOA width must be greater than zero), but the size of the ZOA makes no difference once the swarm is formed.
- The larger the ZOO the wider the swarm distribution, since individuals can influence the direction of the swarm whilst remaining relatively widely dispersed.
- The ZOR is not necessary for a swarm to evolve. It merely controls the minimum distance between model fish. However, in physically embodied experiments (using robots or fish), it is clear that a ZOR is necessary to reduce the chance of collisions.

The average Euclidean distance to the modelfish was used as the fitness function, which proved effective in evolving individuals which were similar to the modelfish in their behaviour, and had similar values for ZOO and ZOR. The main reason for the difference in these parameters was the simplicity of the fitness measure, which only used the distances between the robofish and the modelfish and made no reference to the distance of modelfish *to each other*. If this technique were to be used with real fish, it would be possible to do this using computer vision techniques, although issues with sampling time might arise.

We suggest that if the measure was changed to use the distance the modelfish are from each other as the target distance rather than simply trying to minimise the mean distance to modelfish, then a more accurate estimation of ZOO and ZOR could be expected. This formulation would penalise robofish that swim too closely to the modelfish as well as those which swim too far away from them. In addition, the ZOA was not well estimated in the framework we devised, but we suggest that this too could be addressed by basing a fitness measure that included the period of shoal formation, rather than basing fitness solely on measures of the swarm after it has formed. Other measures such as the spatial point process measure C (Getis and Boots, 1978), which compares an individual's distance to its nearest neighbours against a set of randomly determined points in the world space, could also prove to be effective in developing a fitness function that could recover the model parameters.

In addition to a more sophisticated fitness function, other work could focus on exploring the model more fully. The

results gathered from these experiments could vary greatly depending upon the number of robofish used, the time at which the robofish are introduced (whether a swarm has already formed at that point) or the total size of the swarm. These variables would change the interactions between fish and robofish in different ways, potentially producing wildly divergent emergent effects. The results obtained by these experiments would allow swarming behaviour to be understood more fully, potentially providing valuable insights into the way swarm models are designed and configured.

Another clear area of further work is to look into whether this process can be applied to different models of swarming behaviour. There is little reason to suspect it would fail with the similar models from Aoki and Reynolds (although if this were the case it would raise interesting questions about the nature of these models and the extent of their differences), but adapting it to work on more varied models such as ants, termite mounds or wasp nests may prove more challenging and explore the process of extracting parameters more fully (as well as exploring the individual models being tested).

Of course if these methods are ever used on real life swarms it is unlikely that they will correspond exactly to the models proposed so far. By incorporating the ability for the algorithm to modify the model itself (rather than just the parameters of the model), the process could become far more robust at finding an accurate model of behaviour. This could be done by either switching from a genetic algorithm to genetic programming or by changing to genome representation which allows the model to be altered (such as changing the meaning of zones and limiting the number of neighbours looked amongst others). The increase in flexibility this would produce would allow the process to be applied to far more problems, perhaps allowing novel swarm models to emerge via genetic processes.

Acknowledgments

The authors thank A. Jamie Wood for provision of source code that formed the basis of these experiments. Simon Hickinbotham is funded by the Plazmid project, EPSRC grant EP/F031033/1.

References

- I. Aoki. A simulation study on the schooling mechanism in fish. *Bulletin of the Japanese Society of Scientific Fisheries*, 48(8):1081–1088, 1982.
- R.A. Brooks. *Cambrian Intelligence*. MIT Press, 1999.
- A. Butterfield, V. Vedagiri, E. Lang, C. Lawrence, M. Wakefield, A. Isaev, and G. Huttley. PyEvolve: a toolkit for statistical modelling of molecular evolution. *BMC Bioinformatics*, 5(1):1+, January 2004.
- R. D. Coates and S. J. Hickinbotham. Estimating swarm parameters by evolutionary learning. In *GECCO 2011*. ACM, 2011.
- J. Conley. Evolving boids: Using a genetic algorithm to develop boid behaviors. In *GeoComputation*, 2005.
- I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks. Collective memory and spatial sorting in animal groups. *J. Theor. Biol.*, 218:1–11, 2002.
- J.J. Faria, J.R.G. Dyer, R.O. Clment, I.D. Couzin, N Holt, A.J.W. Ward, D. Waters, and J. Krause. A novel method for investigating the collective behaviour of fish: introducing 'robofish'. *Behav. Ecol. and Sociobiol.*, 64(8): 1211–1218, 2010.
- A. Getis and B. N. Boots. *Models of spatial processes : an approach to the study of point, line, and area patterns*. Cambridge University Press, 1978.
- D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., 1989.
- J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *ICNN*, pages 1942–1948. IEEE, 1995.
- N. Khemka, G. Hushlak, and C. Jacob. Interactive evolutionary evaluation through spatial partitioning of fitness zones. In *EvoWorkshops '09*, pages 432–441. Springer-Verlag, 2009.
- H. Kwong and C. Jacob. Evolutionary exploration of dynamic swarm behaviour. In *CEC '03*, pages 367–374. IEEE, 2003.
- J. Macgill and S. Openshaw. The use of flocks to drive a geographic analysis machine. In *GeoComputation*, 1998.
- C. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Comp. Graphics*, 21(4), 1987.
- D. Stoops, H. Wang, G. Moore, and Y. Bi. Behavioural rule discovery from swarm systems. In *KSEM*, volume 6291 of *LNCS*, pages 506–517. Springer, 2010.
- A. J. J. Wood and G. J. J. Ackland. Evolving the selfish herd: emergence of distinct aggregating strategies in an individual-based model. *Proc. Roy. Soc. B*, 274:1637–1642, May 2007.