

Chapter 2

Information Propagation in a Social Network: The Case of a Fish Schooling Algorithm

A. Brabazon, W. Cui and M. O'Neill

Abstract The propagation of information about the environment amongst animals via social communication has attracted increasing research interest in recent decades with the realisation that many animal species engage in subtle forms of information transfer which had previously escaped notice. From an evolutionary perspective, the widespread existence of social communication mechanisms is not surprising given the significant benefits which can accrue to behaviours such as sharing of information on resources and on environmental threats. More generally, we can consider this process as information flowing between a network of nodes or agents, wherein each agent receives inputs from their senses, processes this information, and in turn through their resulting actions, can influence subsequent actions of other agents. Social communication mechanisms of organisms have inspired the development of several powerful families of optimization algorithms including ant colony optimization and honey bee optimization algorithms. One interesting example of information propagation is provided by the shoaling and schooling behaviours of fish. In this chapter we develop an optimization algorithm (the *Fish Algorithm*) which is inspired by the schooling behaviour of 'golden shiner' fish (*Notemigonus crysoleucas*) and explore the relative importance of social information propagation and individual perception mechanisms in explaining the resulting performance of the algorithm.

A. Brabazon (✉) · W. Cui · M. O'Neill
Complex Adaptive Systems Laboratory and School of Business,
University College Dublin, Dublin, Ireland
e-mail: anthony.brabazon@ucd.ie

W. Cui
e-mail: wei.cui.ireland@gmail.com

M. O'Neill
e-mail: m.oneill@ucd.ie

2.1 Introduction

Swarm behaviour has long attracted research attention with the ‘flocking’ (‘boids’) simulation by Reynolds [31], which mimicked the flocking behaviour of birds, being one of the earliest and best-known examples of such work. In these simulations the flock has no leader (no global control) and co-ordinated movement emerges from the local interactions of individuals in the population. The simulation embeds a few simple rules whereby individuals move in the same direction as their neighbours, remain close to their neighbours, and avoid collisions with their neighbours (producing *alignment, cohesion and separation*). The key characteristic is that each agent only needs local information when deciding how to adjust their movements and yet this, allied to the three simple rules, is sufficient to ensure globally-coordinated behaviour at flock level.

More recently, mechanisms of collective intelligence and their application as practical problem-solving tools, has attracted considerable research interest leading to the development of several families of swarm-inspired algorithms including, ant-colony optimization [7, 9–12], particle swarm optimization [13, 17, 18], bacterial foraging optimization algorithms [28, 29], honey bee algorithms [8, 23, 30, 41], and a developing literature on fish school algorithms. A critical aspect of all of these algorithms is that powerful, emergent, problem-solving occurs as a result of the propagation or sharing of information among a network of individuals, where each individual only possesses local information. Typically the algorithms emphasise the importance of sensing and of communication processes between the agents, and this leads in turn to a discussion of what the agents ‘know’ and how information is propagated or ‘spread’ between individual nodes or agents in the population.

2.1.1 Fish Schooling

Biologists draw an important distinction between dispersion and aggregation economies. In a dispersion economy an increase in group size is correlated with a decrease in the fitness of individual group members, so maximal welfare is obtained when individuals are dispersed and solitary. In contrast, aggregation economies emphasise how group membership can increase the survival rate of individuals particularly when population density is low. A particular example of an aggregation economy is exhibited by some social species of fish which ‘shoal’. ‘Shoaling behaviour’ occurs when fish are observed to cluster together. If the fish also demonstrate a tendency to swim in the same direction in a coordinated manner they are said to ‘school’. These behaviours are common. Approximately a quarter of fish shoal for their entire lives (‘obligate shoalers’ such as tuna, herrings and anchovy) and approximately half shoal for at least part of their lives (‘facultative shoalers’ such as Atlantic cod). More than 4,000 species of pelagic fish are known to be schooling [33] and fish aggregations can be very large with Parrish et al. [27] noting that herring can form schools of a billion or more fish.

Fish shoal and school for mutual protection and to synergistically achieve certain tasks [6]. The benefits include defence against predators as the shoal possesses ‘many eyes’ (or distributed sensing) and has a high-level of vigilance. There is also better protection from individual capture by predators due to the *predator confusion effect* (the many moving targets overloads the predator’s visual channel). The shoal may exhibit enhanced foraging success as many eyes search for food and information on food finds is transmitted through the shoal as the fish can visually monitor each other’s behaviour. Another claimed benefit of schooling is increased hydrodynamic efficiency as the school moves through the water [33].

Schooling behaviour may also reduce or even eliminate the need for sleep. During waking, the brain of most vertebrates is busy processing sensory information, particularly visual information, and this conflicts with the need to refresh and consolidate memories [22]. During schooling, the need for sensory processing, particularly by fish inside the school, is greatly lessened and the burden of sensory processing is shifted from individuals to the entire school [22]. Schooling behaviours may therefore play a role similar to that of restful waking or sleep in non-schooling fish species.

2.1.2 How Do Fish Schools Make Decisions?

A natural question facing any modeller who is seeking to develop an optimization algorithm using fish school inspired behaviours is how do fish schools actually make decisions—and critically, is there any theoretical reason to suppose that distributed sensing can generate a more ‘intelligent’ decision than the decision that could be made by an individual fish?

When we consider the dynamic environment which faces a school of fish, it is apparent that many complex decisions are faced. In which direction should it swim if faced by a predator? When should it stop and forage? When and where should it migrate? In contrast to mammal herds, fish schools have no leader. Each of the fish in a school has similar sensing capabilities and similar behaviour patterns for acting on sensory information [33] but there is no strong evidence that individual fish can undertake highly complex information processing.

A recent study [35] has suggested that fish schools may implement a form of consensus-based decision making employing a simple quorum rule. Under a quorum rule, an individual’s probability of committing to a particular decision option increases sharply when a threshold number of other individuals have committed to it. Hence, if individuals can observe the decisions of others before committing themselves to a decision such as what direction in which to swim, a relatively naive copying behaviour can be an effective strategy for successful decision making, without the need for individuals to undertake complicated information processing.

Distributed perception and quorum decision processes combine therefore to create a form of collective intelligence which can reduce the need to undertake complex cognition at agent level, and can also allow robust decision making to take place even when individual perceptions are noisy. The quality of the decision and the size of the group are highly correlated [35] so the quality of the decision increases as

group size increases. This suggests that fish school behaviours can indeed form a useful platform for the development of optimization algorithms. In this study we propose an optimization algorithm inspired by a recent study by Berdahl et al. [3] of golden shiner fish and within this framework, explore the relative importance of social information propagation and individual perception mechanisms in explaining the resulting performance of the algorithm.

The remainder of this contribution is organised as follows. Section 2.2 provides some background literature on previous work which has adopted a fish school metaphor in the development of optimization algorithms and on the specific biological model underlying this study. Section 2.3 describes the proposed algorithm (termed the ‘Fish Algorithm’). The results from a series of test problems are provided in Sect. 2.5 and finally, conclusions and opportunities for future work are discussed in Sect. 2.6.

2.2 Background

A number of previous studies have previously employed a fish school metaphor to develop algorithms for optimization and clustering ([1, 2, 16, 19, 38, 44] provide a sampling of this work). Two of the better-known approaches are Fish School Search (FSS) [2] and the Artificial Fish Swarm Algorithm (AFSA) [19].

In FSS the algorithm implements three fish behaviours, namely feeding, swimming and breeding. The behaviour of feeding is inspired by the natural instinct of fishes to feed, feeding here is a metaphor for the evaluation of candidate solutions in the search space; the swimming behaviour aims at mimicking the coordinated movement of fish in a school guiding the search process; the breeding behaviour is inspired by natural selection a metaphor for exploitation of better-adapted candidate solutions. The fish (agents) swim (search) for food (candidate solutions) in an aquarium (search space) and the weight of each fish acts as an innate memory of its past individual success. Unlike particle swarm optimization (PSO) [17, 18], no direct memory of a personal best location or a global best location is maintained. FSS has shown itself to be a powerful optimization algorithm demonstrating good results on a range of optimization problems.

The AFSA [19] embeds a number of fish behaviours including preying, swarming, and following so that the behaviour of an artificial fish depends on its its current state, its local environmental state (including the quality of its current location and the states of nearby companions). A good review of the recent literature on AFSA is provided in [24].

2.2.1 Application of Fish School Algorithms

Fish school algorithms have been applied for a wide variety of applications and an excellent overview of these is provided by [24]. Canonical versions of fish school

algorithms typically employ a real-valued representation and are used to search in an environment/problem space for a ‘point’ which corresponds to an optimal solution vector (a simple exemplar would be a vector of parameters for a mathematical model which is being calibrated using a training dataset). Hence, the algorithms can be applied to any real-valued optimization problem. The canonical algorithms can also be modified for application to discrete optimization, multi-objective optimization and clustering.

A sampling of the applications for which fish school algorithms have been employed include, the determination of the optimal deployment strategy for nodes in a wireless network [4, 42]; the optimal deployment of directional visible light sensor networks for battlefield surveillance and intrusion detection [43]; road traffic network design [21]; the optimization of weights in a feed-forward neural network model [40]; quality of service (QoS) graded optimization in electric power communication networks [25]; the optimization of the parameters of membership functions for a fuzzy logic controller [36]; task scheduling in a multi robot group [37]; aircraft landing scheduling in a multi-runway airport [5]; and efficient job scheduling in grid computing [14].

2.2.2 Golden Shiner Fish

A practical issue that arises in attempting to develop an algorithm based on the behaviour of fish schools is that we have relatively little hard data on the behavioural mechanisms which underlie schooling phenomena. At the level of the individual, agents respond to their own sensory inputs, physiological and cognitive states, and locomotory constraints [15] and it is not trivial to disentangle the relative influence of each of these. At group-level, it is often difficult to experimentally observe the mechanics of the movement of animal groups or fish schools, and hence much previous work developing fish school algorithms has relied on high-level observations of fish behaviour rather than on granular empirical data on these behaviours.

In this study we draw inspiration from a detailed study of the behaviour of a species of schooling fish ‘golden shiners’ which display a marked preference for shaded habitat [3]. These fish are strongly social and form shoals of some 200–250 individuals in the fresh-water lakes where they live.

In order to investigate the mechanism underlying the observed collective response of golden shiner fish to light gradients, fish were tracked individually to obtain information on individual and group trajectories. The study examined the degree to which the motion of individuals is explained by individual perception (steepest direction of light gradient as seen by the individual fish) and social influences based on distributed perception (positions of conspecifics). The results indicated that an individual’s acceleration was more influenced by the location of conspecifics than by locally-perceived environmental gradients. When the magnitude of the social vector was high (all conspecifics moving in similar direction) the social influence was dominant. As noted by [32], all forms of animal communication are closely tied

to the senses. In the case of fish, visual cues form the primary basis of the social communication mechanism as schooling fish are able to observe the movements of their neighbouring conspecifics.

2.3 Fish Algorithm

An important question that underlies the design of foraging strategies, or the design of optimization algorithms, is what is the most effective way of searching for objects whose location is not known a priori. In foraging, the search could be guided by external cues, either via past experience (memory) or sensory inputs (such as vision) of the searcher. Alternatively, the search process could be stochastic (i.e. undirected). When the location of the target objects is unknown, a degree of ‘guessing’ is unavoidable, and probabilistic or stochastic strategies are required [39].

In the proposed algorithm, the movement of each fish is governed by three biologically-inspired factors which are described below, and also embeds a stochastic element. In each iteration of the algorithm, a fish is displaced from its previous position through the application of a velocity vector:

$$p_{i,t} = p_{i,t-1} + v_{i,t} \quad (2.1)$$

where $p_{i,t}$ is the position of the i th fish at current iteration of the algorithm (t), $p_{i,t-1}$ is the position of the i th fish at the previous iteration ($t - 1$), and $v_{i,t}$ is its velocity.

The velocity update is a composite of three elements, prior period velocity, an individual perception mechanism, and social influence via the distributed perception of conspecifics. The update is:

$$v_{i,t} = v_{i,t-1} + DP_{i,t} + IP_{i,t} \quad (2.2)$$

or more generally

$$v_{i,t} = w_1 v_{i,t-1} + w_2 DP_{i,t} + w_3 IP_{i,t} \quad (2.3)$$

The difference between the two update equations is that weight coefficients are given to each of the update items in Eq. 2.3. In all the experiments of this study, Eq. 2.2 is used for velocity update. While the form of the velocity update bears a passing resemblance to the standard PSO velocity update, in that both have three terms, it should be noted that the operationalisation of the individual perception and distributed perception mechanisms is completely different to the memory-based concepts of $pbest$ and $gbest$ in PSO. The next subsection explains the operation of the two perception mechanisms.

2.3.1 Prior Period Velocity

The inclusion of a prior period velocity can be considered as a proxy for momentum or inertia. Although this feature was not described in the study of golden shiner fish [3],

the inclusion of this term is motivated by empirical evidence from the movement ecology literature which indicates that organisms tend to move with a ‘directional persistence’ [39].

2.3.2 Distributed Perception Influence

In all social models, a key element is how the overall population influences the decisions of each agent at each time step. Typically, the actions of each agent are influenced by a subset of the population who are within an ‘interaction range’ of them. This influence can be modelled in a variety of ways including the fraction of an individual’s neighbours taking a particular course of action or the action of their nearest neighbour. In this study we model the distributed perception influence for the i th fish by the following:

$$DP_i = \frac{\sum_{j=1}^{N_i^{DP}} (p_j - p_i)}{N_i^{DP}}, \quad j \neq i \tag{2.4}$$

where p_i is the position of the i th fish, and the sum is calculated over all neighbours within an assumed range of interaction of the i th fish r_{DP} , that is $0 < |p_j - p_i| \leq r_{DP}$, where p_j is the position of the j th neighbouring fish, and N_i^{DP} is number of neighbours in the assumed range of interaction of the i th fish. If there are no neighbours in its assumed range of interaction, this term becomes zero. Figure 2.1 shows how the i th fish is affected by the three neighbouring fish (p_1, p_2, p_3) which are within its visible range (defined by the radius r_{DP}).

Alternative methods of modelling this social influence could be implemented such as only considering neighbours within the angular visual range of each agent as suggested by Miller et al. [26]. While this would be more plausible from a biological perspective, it would impose additional computational complexity so we use a simpler approach in this chapter which implicitly assumes 360° vision. Note that

Fig. 2.1 Illustration of distributed perception

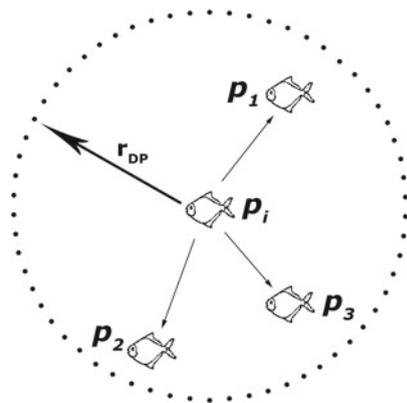
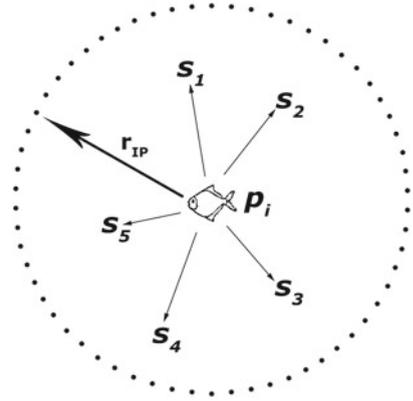


Fig. 2.2 Illustration of individual perception



in this mechanism, no direct account is taken of the light gradient in any direction by an individual fish, rather the influence on the movement of a fish is completely determined by the movement of its neighbours.

2.3.3 Individual Perception Influence

Individual perception is implemented as follows. At each update, each fish assesses the local ‘light’ gradient surrounding it, by drawing N_i^{IP} samples within an assumed ‘visibility’ region of radius r_{IP} . While a real-world fish will have a specific angle of vision depending on its own body structure, we adopt a random sampling in a hypersphere around the fish on grounds of generality. The individual perception influence for the i th fish is determined by:

$$IP_i = \frac{\sum_{j=1}^{N_i^{IP}} (s_j - p_i) * fit_j}{\sum_{j=1}^{N_i^{IP}} fit_j}, \quad j \neq i \quad (2.5)$$

where p_i is the position of the i th fish, r_{IP} is the radius of the assumed range within which the i th fish can sense environmental information, N_i^{IP} is the number of samples which the i th fish generates, s_j is the position of the j th sample ($0 < |s_j - p_i| \leq r_{IP}$), and fit_j is the fitness value (or ‘quality’) of the j th sample. Figure 2.2 demonstrates how the i th fish is influenced by the five random samples ($s_1 - s_5$) in the perception range with a radius r_{IP} .

2.4 Experimental Design

In this section we describe the test functions used in all our experiments, we outline the precise experiments undertaken in this study, and we describe the associated experimental parameters.

2.4.1 Benchmark Functions

Twelve standard benchmark problems (outlined in Table 2.1) taken from the optimization literature were used to test the developed algorithms. All problems are examined at two levels of dimensionality, namely 30 and 60 dimensions. The aim in all the experiments is to find the vector of values which minimises the value of a test function, hence, we can define the fitness of a solution vector as the value of the test function at that location, with lower values (in this case, as we are minimising) indicating a better quality (or ‘fitter’) solution.

Two of the functions namely, the Sphere and Rosenbrock functions, represent unimodal problems. The Griewank and Rastrigin functions are more complex and contain multiple local optima. In following paragraphs, we provide a brief description of these test functions in order to provide some intuition as to their structure.

The last six problems are drawn from the optimization benchmark functions used in the IEEE CEC 2005 Special Session on Real-Parameter Optimization [34]. An interesting aspect of these functions is that the global optima are shifted or rotated (shift is given by the parameter \mathbf{o} , and the parameter M represents an orthogonal matrix which is used to rotate the function). The net effect of these processes is to

Table 2.1 Twelve optimization problems

Name	Function	Search space	Optima
Sphere	$F_1(\mathbf{x}) = \sum_{i=1}^n x_i^2$	$[-3.12 \ 7.12]^D$	0
Rosenbrock	$F_2(\mathbf{x}) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	$[-30 \ 30]^D$	0
Ackley	$F_3(\mathbf{x}) = -20 \exp\left(-0.2\sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D}\sqrt{\sum_{i=1}^D \cos(2\pi x_i)}\right) + 20 + e$	$[-32.768 \ 32.768]^D$	0
Griewank	$F_4(\mathbf{x}) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right)$	$[-600 \ 600]^D$	0
Rastrigin	$F_5(\mathbf{x}) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$	$[-5.12 \ 5.12]^D$	0
Schwefel	$F_6(\mathbf{x}) = 418.9829 \times D - \sum_{i=1}^D x_i \sin(x_i ^{\frac{1}{2}})$	$[-500 \ 500]^D$	0
Shifted sphere	$F_7(\mathbf{x}) = \sum_{i=1}^D z_i^2 - 450, \mathbf{z} = \mathbf{x} - \mathbf{o}$	$[-100 \ 100]^D$	-450
Shifted rosenbrock	$F_8(\mathbf{x}) = \sum_{i=1}^{D-1} 100(z_i^2 - z_{i+1})^2 + (x_i - 1)^2 + 390, \mathbf{z} = \mathbf{x} - \mathbf{o} + 1$	$[-100 \ 100]^D$	390
Shifted rotated ackley	$F_9(\mathbf{x}) = -20 \exp(-0.2\sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}) - \exp(\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i)) + 20 + e - 140, \mathbf{z} = (\mathbf{x} - \mathbf{o}) * M$	$[-32 \ 32]^D$	-140
Shifted rotated griewank	$F_{10}(\mathbf{x}) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 - 180, \mathbf{z} = (\mathbf{x} - \mathbf{o}) * M$	$[-600 \ 600]^D$	-180
Shifted rotated rastrigin	$F_{11}(\mathbf{x}) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) - 330, \mathbf{z} = (\mathbf{x} - \mathbf{o}) * M$	$[-5 \ 5]^D$	-310
Shifted schwefel	$F_{12}(\mathbf{x}) = \sum_{i=1}^D (\sum_{j=1}^i z_j)^2 - 450, \mathbf{z} = \mathbf{x} - \mathbf{o}$	$[-100 \ 100]^D$	-450

move the global optimum away from the origin in each case, due to the known issues with using standard, benchmark functions which have their optimum at the origin [20]. These issues can sometimes be exploited by algorithms to produce an upward bias in reported performance. Problems include the fact that,

1. many popular benchmark functions are symmetric, and hence have the same optimal parameter values for all dimensions (for example, a vector of zeros); and
2. the global optimum may lie at the centre of the search space (this can produce problems if search agents are initialised randomly along the range of each dimension).

Hence, considering the conventional sphere function,

$$f(x) = \sum_{i=1}^D x_i^2$$

the shifted sphere function is given by:

$$f(x) = \sum_{i=1}^D (x_i - o_i)^2$$

and the shifted rotated sphere function is given by:

$$f(x) = \sum_{i=1}^D [(x_i - o_i) * M]^2$$

2.4.1.1 Sphere Function

This is a relatively simple test function as it is continuous, convex and unimodal. The function is defined as $\sum_{i=1}^n x_i^2$. In Fig. 2.3, n is set to 2 for ease of illustration, and $-5.12 \leq x_i \leq 5.12$. The objective is to find the values of x_1 and x_2 which minimise the value of the function. By inspection, the global minimum (zero) occurs when x_1 and x_2 are zero. While we illustrate the function here for the case where there are two inputs, in our experiments on each test function we undertake a search for the global optimum in both 30 and 60 dimensions.

2.4.1.2 Griewangk's Function

Griewangk's function has many local minima in the region of the global minimum, with these minima being regularly distributed. The presence of many local minima renders the determination of the optimal value for this function more difficult than is the case for the Sphere function. The function is defined as:

$$F(x) = 1 + \sum_{i=1}^n \left[\frac{x_i^2}{4000} \right] - \prod_{i=1}^n \left[\cos \left(\frac{x_i}{\sqrt{i}} \right) \right] \quad (2.6)$$

Fig. 2.3 Sphere function

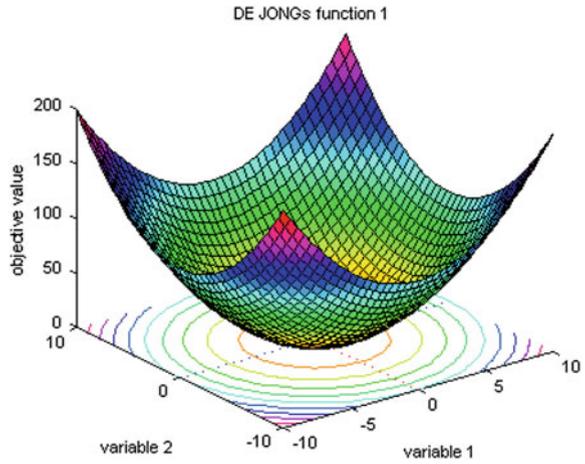
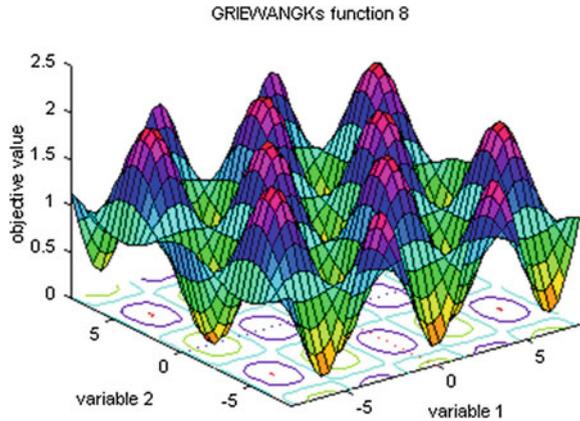


Fig. 2.4 Griewangk's function, range +/- 5



where $n = 2$ (in Fig. 2.4), and $-600 \leq x_i \leq 600$. The global minimum (zero) occurs when all x_i are 0.

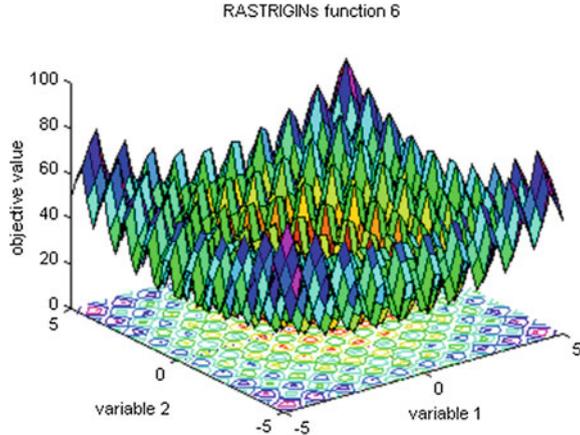
2.4.1.3 Rastrigin's Function

Rastrigin's function has a cosine modulation to produce many local minima (Fig. 2.5). This produces a test function which is highly multimodal. However, the location of the minima are regularly distributed. The function is defined as:

$$F(x) = n * A + \sum_{i=1}^n \left[x_i^2 - A * \cos(2\pi x_i) \right] \tag{2.7}$$

with $A = 10$ and $n = 2$ (in the illustration) and $-5.12 \leq x_i \leq 5.12$. The global minimum (zero) occurs when all x_i are zero.

Fig. 2.5 Rastrigin's function



The Rosenbrock function (also known as Rosenbrock's valley or Rosenbrock's banana function) is a non-convex function. The global minimum is inside a long, narrow, parabolic shaped flat valley. While it is relatively easy to find the valley, it is difficult to find the global optimum point within this.

2.4.2 Experiments

Two groups of experiments are undertaken. Initially, we determine the performance of the canonical fish algorithm (denoted as 'FA') which uses the velocity update described in Eq. 2.2, on all the test problems. Next we develop three variants of the canonical FA which switch off, in turn, the momentum, the distributed perception (DP) and the individual perception (IP) influences (these algorithmic variants are denoted as FA1, FA2 and FA3 respectively). The performance of each of these variants on the test problems is examined in order to gain insight into the role that each of the three components of the velocity update step plays in determining the FA's overall performance.

The second set of experiments examines the sensitivity of the canonical FA to changes in two of its parameters, namely the radius of perception in both r_{DP} & r_{IP} , and the number of samples (denoted as s) used in the simulated individual perception (IP) component. The chosen values of these parameters are shown in Table 2.2.

From a biological point of view, it is plausible to assume that fish have a bigger radius for DP than IP, namely $r_{DP} > r_{IP}$. The value chosen for the two radii is problem specific, as it is influenced by the choice of the number of fish (N), the radius (size) of the search space (R) and the dimensionality of the this space (D). In the FA algorithm, the values of r_{DP} and r_{IP} were chosen after initial experimentation as $\frac{R}{1.5\sqrt[3]{N}}$ and $\frac{R}{1.8\sqrt[3]{N}}$ so that in most cases each fish has neighbouring fish within the radius r_{DP} .

Table 2.2 Parameter setting of algorithms

Algorithm	Radius of DP (r_{DP})	Radius of IP (r_{IP})	Number of samples in IP (s)	Velocity updating equation
FA	$\frac{R}{1.5\sqrt[D]{N}}$	$\frac{R}{1.8\sqrt[D]{N}}$	5	$v_{i,t} = v_{i,t-1} + DP_{i,t} + IP_{i,t}$
FA1	$\frac{R}{1.5\sqrt[D]{N}}$	$\frac{R}{1.8\sqrt[D]{N}}$	5	$v_{i,t} = 0 + DP_{i,t} + IP_{i,t}$
FA2	$\frac{R}{1.5\sqrt[D]{N}}$	$\frac{R}{1.8\sqrt[D]{N}}$	5	$v_{i,t} = v_{i,t-1} + 0 + IP_{i,t}$
FA3	$\frac{R}{1.5\sqrt[D]{N}}$	$\frac{R}{1.8\sqrt[D]{N}}$	5	$v_{i,t} = v_{i,t-1} + DP_{i,t} + 0$
FAa	$\frac{R}{3\sqrt[D]{N}}$	$\frac{R}{3.6\sqrt[D]{N}}$	5	$v_{i,t} = v_{i,t-1} + DP_{i,t} + IP_{i,t}$
FAb	$\frac{R}{1\sqrt[D]{N}}$	$\frac{R}{1\sqrt[D]{N}}$	5	$v_{i,t} = v_{i,t-1} + DP_{i,t} + IP_{i,t}$
FAC	$\frac{R}{1.5\sqrt[D]{N}}$	$\frac{R}{1.8\sqrt[D]{N}}$	10	$v_{i,t} = v_{i,t-1} + DP_{i,t} + IP_{i,t}$
FAd	$\frac{R}{1.5\sqrt[D]{N}}$	$\frac{R}{1.8\sqrt[D]{N}}$	1	$v_{i,t} = v_{i,t-1} + DP_{i,t} + IP_{i,t}$

Note R is the radius of the search space

D is the dimension of the test problem

N is the number of fish

In order to undertake some sensitivity analysis, four variants of the FA algorithm are developed. In the FAa algorithm, the values of r_{DP} and r_{IP} are set to be half of those in the FA algorithm. In the FAb algorithm, the values of r_{DP} and r_{IP} are set to be larger than those in the FA algorithm. In the FAC algorithm, the value of s is increased to 10 (as against 5 in the FA algorithm). In the FAd algorithm, the value of s is reduced to 1. Note that in these latter two cases, the effect is to alter the implicit weighting accorded to the IP mechanism in the velocity update step, as in all our experiments, each algorithmic variant is accorded the same number of function evaluations.

We note that in this study the focus is not on designing the ‘best’ possible variant of the fish algorithm for optimization purposes. Rather, using the framework outlined in Sect. 2.3 we seek to examine the relative importance of social information propagation and individual perception mechanisms in explaining the resulting performance of the algorithm. We also wish to examine the sensitivity of the performance to changes in key parameters in each mechanism (range of perception and relative weight placed on IP vs. DP).

2.4.3 Experimental Settings

Table 2.3 describes the parameter settings adopted. In each experiment, 40 fish are used. All reported results are averaged over 30 runs and we test the statistical significance of all differences in the means using a t -test. In all experiments, an equivalent number of function evaluations are undertaken in order to ensure a fair comparison between the different algorithms. The experiments were undertaken on an Intel Core i7 (2.93 GHz) system with 12 GB RAM.

Table 2.3 Parameter setting of experiments

Parameters	Values
Trials	30
Size of fish school	$N = 40$
Dimension of problem	$D = 30, 60$

2.5 Results

Tables 2.4, 2.5, 2.6 and 2.7, and Figs. 2.6 and 2.7 present the results from our experiments. The Tables show for each algorithm variant & test function combination (for both $D = 30$ and $D = 60$), the end of run evaluation for each test function at the best location (solution vector) found across all 30 runs ('Best'), the evaluation of each benchmark function averaged over the best location (solution vector) found on each of the 30 individual runs ('Mean'), and the associated standard deviation over all 30 runs. The Tables also present the results from our statistical testing of a variety of hypotheses. In all cases, low p values indicate that the null hypothesis of 'no difference between the means' is rejected (a 95 % level is applied).

Figures 2.6 and 2.7 illustrate the 'Mean' (defined as above) evaluation of each benchmark function and indicate how this value changes (improves) as the number of iterations increases (only the $D = 60$ case is shown in order to conserve space).

2.5.1 Hypotheses Examined

In order to facilitate interpretation of the statistical tests we outline the notation used below.

The first set of hypotheses concern the testing of the importance of each component of the fish algorithm (FA). The null hypothesis is that there is no difference in the performance (i.e. 'Mean') between the algorithm with a component turned off and the canonical FA. Therefore three hypotheses are tested as follows.

- H_1 : no difference in performance between the FA and the FA1 algorithm;
- H_2 : no difference in performance between the FA and the FA2 algorithm;
- H_3 : no difference in performance between the FA and the FA3 algorithm.

The next set of hypotheses concern the analysis of differing parameter settings for FA. Four cases are examined, FAa, FAb, FAc and FAd and the relevant hypotheses are denoted as follows.

- H_a : no difference in performance between the FA and the FAa algorithm;
- H_b : no difference in performance between the FA and the FAb algorithm;
- H_c : no difference in performance between the FA and the FAc algorithm;
- H_d : no difference in performance between the FA and the FAd algorithm.

Table 2.4 End of run results for each algorithmic variant for F1–F6 (30D case)

Algorithm		Function 1	Function 2	Function 3	Function 4	Function 5	Function 6
FA	Best	23.20	5,711,602	4.75	92.64	1269.10	8572.38
	Mean	31.60	10,997,353	5.18	110.91	1605.36	9109.52
	Std	4.09	1,809,538	0.15	9.76	123.22	217.68
FAa	Best	116.19	126,828,764	8.11	423.36	5755.66	7093.78
	Mean	160.83	229,847,840	9.03	550.35	7076.47	7563.07
	Std	17.70	38,217,754	0.29	57.36	606.75	263.16
	H_a	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
FAb	Best	22.95	2,941,104	4.63	68.27	1141.99	8476.14
	Mean	28.31	7,045,931	5.06	96.18	1426.84	9228.50
	Std	2.15	1,817,708	0.15	9.30	105.46	193.53
	H_b	0.0003	0.0000	0.0035	0.0000	0.0000	0.0291
FAc	Best	25.47	9,864,494	4.89	100.96	1400.92	8129.12
	Mean	33.90	15,997,277	5.19	119.82	1734.15	9224.45
	Std	3.66	2,195,339	0.13	9.56	135.65	345.81
	H_c	0.0254	0.0000	0.8633	0.0007	0.0003	0.1289
FAd	Best	67.55	8,788,5	7.48	334.16	4275.20	6603.14
	Mean	125.18	130,952,522	8.31	432.63	5439.34	7291.36
	Std	27.59	63,633,167	0.40	65.45	781.83	238.38
	H_d	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
FA1	Best	5.87	769,851	3.24	30.26	605.85	9085.49
	Mean	11.95	2,016,472	3.82	40.37	804.36	9638.35
	Std	2.73	863,653	0.22	7.76	109.39	197.50
	H_1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
FA2	Best	142.02	102,254,451	8.65	391.98	5136.39	6705.87
	Mean	161.94	219,256,368	9.03	537.83	6996.73	7498.80
	Std	11.31	42,204,038	0.19	56.26	553.66	320.17
	H_2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
FA3	Best	54.24	34,303,799	6.69	174.07	2803.55	8402.80
	Mean	118.95	147,475,612	8.08	375.25	4866.90	9358.78
	Std	40.72	76,584,873	0.90	104.84	1748.87	403.90
	H_3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0043

2.5.2 Discussion of Results

Initially we overview Figs. 2.6 and 2.7 to get an idea of the general trends in the results. Taking a high-level perspective, we note that while the performance of each algorithmic variant varies depending on the test function examined, the performance of the canonical version of FA is generally better than that of FA2 or FA3 (which

Table 2.5 End of run results for each algorithmic variant for F7–F12 (30D case)

Algorithm		Function 7	Function 8	Function 9	Function 10	Function 11	Function 12
FA	Best	43609.53	33,251,772,403	-119.15	1961.38	341.18	69551.47
	Mean	73399.67	51,123,866,487	-119.06	3363.09	429.12	153085.92
	Std	7333.86	5,270,435,371	0.03	426.19	35.67	45548.08
FAa	Best	60402.13	36,139,582,367	-119.18	1687.58	273.99	73671.21
	Mean	73453.61	52,326,755,580	-119.06	2343.38	345.95	92338.86
	Std	6757.77	7,917,980,119	0.03	241.66	35.97	11512.57
	H_a	0.9765	0.4913	0.7135	0.0000	0.0000	0.0000
FAb	Best	48692.70	18,887,598,328	-119.19	2137.20	246.97	68000.39
	Mean	56701.66	29,121,822,007	-119.07	2684.83	307.88	123883.58
	Std	4172.39	3,406,108,447	0.05	196.01	29.30	46059.22
	H_b	0.0000	0.0000	0.1841	0.0000	0.0000	0.0165
FAc	Best	63122.53	46,219,215,386	-119.15	2833.47	399.57	69209.61
	Mean	84807.43	65,737,688,501	-119.02	3928.29	518.35	162667.01
	Std	8166.06	11,155,657,744	0.05	462.02	58.75	50988.65
	H_c	0.0000	0.0000	0.0025	0.0000	0.0000	0.4459
FAd	Best	54346.58	11,983,840,968	-119.26	1372.58	162.86	46895.82
	Mean	65107.04	29,184,496,221	-119.14	1788.40	286.61	75499.57
	Std	5908.49	7,029,267,934	0.05	228.13	38.78	9330.60
	H_d	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
FA1	Best	63849.53	24,242,817,446	-119.15	2921.08	356.39	63231.72
	Mean	78639.07	45,195,413,457	-119.06	3887.02	476.55	127709.92
	Std	7724.06	10,278,935,692	0.03	483.17	50.81	45586.37
	H_1	0.0092	0.0067	0.8309	0.0000	0.0001	0.0352
FA2	Best	51504.81	35,277,599,066	-119.10	1802.17	249.84	85342.68
	Mean	71047.83	53,586,077,306	-119.05	2295.75	356.74	99470.47
	Std	8502.51	10,895,945,623	0.03	277.71	44.55	7656.39
	H_2	0.2560	0.2698	0.2514	0.0000	0.0000	0.0000
FA3	Best	60643.40	23,910,071,808	-119.15	2913.74	341.82	74311.54
	Mean	101787.20	83,404,345,263	-118.97	4014.97	573.92	144463.81
	Std	16783.55	27,221,806,065	0.07	567.37	105.95	37769.59
	H_3	0.0000	0.0000	0.0000	0.0000	0.0000	0.4281

have DP and IP turned off respectively), but that FA1 variant (in which momentum is turned off) appears to perform better than FA on several problems. Looking at the three variants FA1–FA3, FA1 performs better than either of the other two variants, with FA2 generally slightly outperforming FA3.

Taking the results together, it appears that DP (distributed perception) and IP (individual perception) contribute usefully to the search process but that the importance of momentum is not clearly demonstrated. It also appears that the DP and IP

Table 2.6 End of run results for each algorithmic variant for F1–F6 (60D case)

Algorithm		Function 1	Function 2	Function 3	Function 4	Function 5	Function 6
FA	Best	287.79	311,582,651	6.10	850.62	4602.89	17392.30
	Mean	379.79	607,913,058	9.21	1305.69	15203.91	18062.21
	Std	40.32	88,975,366	1.05	142.69	2687.78	324.02
FAa	Best	316.91	513,332,990	9.16	1230.04	14316.41	16740.36
	Mean	394.05	631,994,795	9.66	1386.62	16719.04	17920.98
	Std	32.12	64,533,215	0.22	81.55	1105.92	449.75
	H_a	0.1355	0.2350	0.0250	0.0091	0.0060	0.1682
FAb	Best	77.71	26,742,064	5.49	243.21	3860.06	19609.69
	Mean	88.42	36,348,031	5.92	307.41	4223.62	20443.04
	Std	4.12	4,381,860	0.11	18.18	163.05	302.90
	H_b	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
FAc	Best	85.60	43,270,190	5.62	305.97	3916.05	17972.25
	Mean	237.08	449,166,558	7.65	776.13	7813.33	18633.15
	Std	121.05	247,705,930	1.58	459.70	5026.22	387.10
	H_c	0.0000	0.0016	0.0000	0.0000	0.0000	0.0000
FAd	Best	312.57	371,304,375	8.89	1013.70	12571.32	16910.43
	Mean	388.55	589,343,313	9.70	1360.94	16819.07	17670.53
	Std	33.65	104,217,530	0.27	118.10	1412.29	322.42
	H_d	0.2795	0.4060	0.0011	0.0544	0.0003	0.0000
FA1	Best	25.57	2,635,463	3.83	82.36	1444.85	19840.50
	Mean	30.98	4,880,904	4.18	107.38	1847.15	21007.05
	Std	3.58	1,429,855	0.14	11.29	159.25	443.44
	H_1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
FA2	Best	322.44	492,743,476	9.16	1017.49	15048.75	17191.05
	Mean	387.32	622,753,537	9.62	1334.03	16640.72	18069.47
	Std	24.19	69,381,378	0.18	109.47	816.48	392.15
	H_2	0.3843	0.4742	0.0393	0.3916	0.0069	0.9380
FA3	Best	322.44	492,743,476	8.39	961.99	12802.50	19901.43
	Mean	389.43	632,722,577	9.62	1303.44	16563.01	20945.41
	Std	26.05	73,067,155	0.33	137.22	1230.47	416.20
	H_3	0.2764	0.2427	0.0472	0.9508	0.0146	0.0000

mechanisms can produce relatively similar levels of performance by the end of each experiment.

Next, we take a high-level overview of the performance of FA versus the algorithmic variants with different parameter settings (FAa–FAd). As before, the performance of the algorithmic variants depends on the test problem but in general, the ordering of performance (on the 60D cases in the Figures) appears to be $FAb > FAc > FA > FAd > FAa$. This ordering is plausible as fish in the algorithmic

Table 2.7 End of run results for each algorithmic variant for F7–F12 (60D case)

Algorithm		Function 7	Function 8	Function 9	Function 10	Function 11	Function 12
FA	Best	159458.20	101,440,099,772	-118.92	5367.39	1163.32	330672.22
	Mean	210503.03	188,388,985,169	-118.80	7107.06	1420.50	397832.08
	Std	19728.45	32,095,577,831	0.03	726.08	89.72	37187.04
FAa	Best	173831.05	142,801,200,876	-118.92	5655.23	1180.41	284368.86
	Mean	213184.06	212,224,115,674	-118.80	7197.15	1420.27	374998.36
	Std	17464.80	21,633,796,969	0.04	823.27	107.99	44435.06
	H_a	0.5794	0.0013	0.6089	0.6547	0.9928	0.0350
FAb	Best	108596.74	59,065,713,240	-118.88	3658.36	884.23	311357.75
	Mean	130188.82	70,454,662,207	-118.80	5137.79	1079.93	713842.41
	Std	7705.40	6,690,493,140	0.03	686.85	64.09	338861.85
	H_b	0.0000	0.0000	0.9847	0.0000	0.0000	0.0000
FAc	Best	150520.29	106,064,987,767	-118.85	4919.17	1124.88	308881.61
	Mean	181714.55	166,416,739,386	-118.79	7095.28	1330.02	521952.55
	Std	13605.42	36,340,990,005	0.03	1036.29	92.02	108045.25
	H_c	0.0000	0.0160	0.3545	0.9595	0.0003	0.0000
FAd	Best	192437.89	125,451,253,553	-118.93	5592.48	1231.55	239554.04
	Mean	218978.72	190,448,882,676	-118.86	7100.64	1426.49	296261.74
	Std	12060.85	27,148,876,846	0.03	662.18	89.36	27396.78
	H_d	0.0134	0.7504	0.0000	0.9666	0.7655	0.0000
FA1	Best	146554.32	66,996,750,338	-118.89	6427.89	1059.54	351171.76
	Mean	162332.97	86,987,862,848	-118.80	7144.74	1162.22	864918.75
	Std	9607.24	11,310,775,223	0.03	404.89	57.04	346570.48
	H_1	0.0000	0.0000	0.8134	0.8048	0.0000	0.0000
FA2	Best	187090.19	140,251,557,412	-118.85	5772.94	1132.50	301394.25
	Mean	211949.44	200,622,613,125	-118.79	7009.69	1404.10	393418.16
	Std	11137.19	21,475,434,745	0.02	705.40	111.61	46386.99
	H_2	0.7278	0.0880	0.6529	0.6003	0.5329	0.6858
FA3	Best	197106.30	192,083,954,497	-118.82	6664.71	1300.24	314263.05
	Mean	251792.25	274,617,244,480	-118.69	9179.32	1707.49	664776.88
	Std	20690.26	44,967,767,801	0.03	1189.09	157.68	193674.39
	H_3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

variant FAb have a wider ‘perception radius’ than do the fish in any of the other algorithmic variants allowing them to perceive information from a greater volume of the search space. Conversely, the relatively poorer search performance of FAa is not unexpected as it has a smaller perception radius than the other algorithm variants.

Hence, from a high-level overview of Figs. 2.6 and 2.7, the key points are that while IP and DP provide useful information for the search process, the momentum mechanism does not appear to be as important. It is also evident that the performance

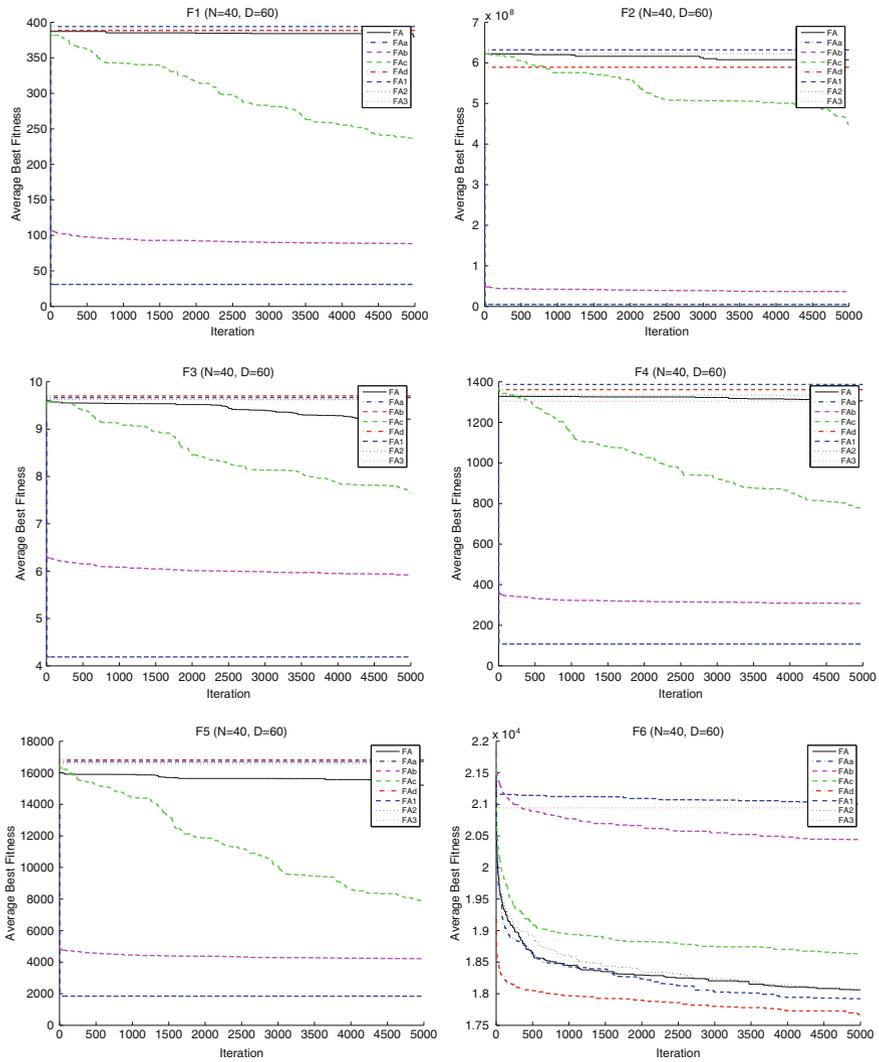


Fig. 2.6 Average best performance (averaged over 30 trials) of each algorithm variant on test problems F1–F6 (60D)

of the algorithm is sensitive to choices of perception radius, with increases in this parameter leading to enhanced performance.

Next, we proceed to look at the results in Tables 2.4, 2.5, 2.6 and 2.7 in order to obtain finer detail.

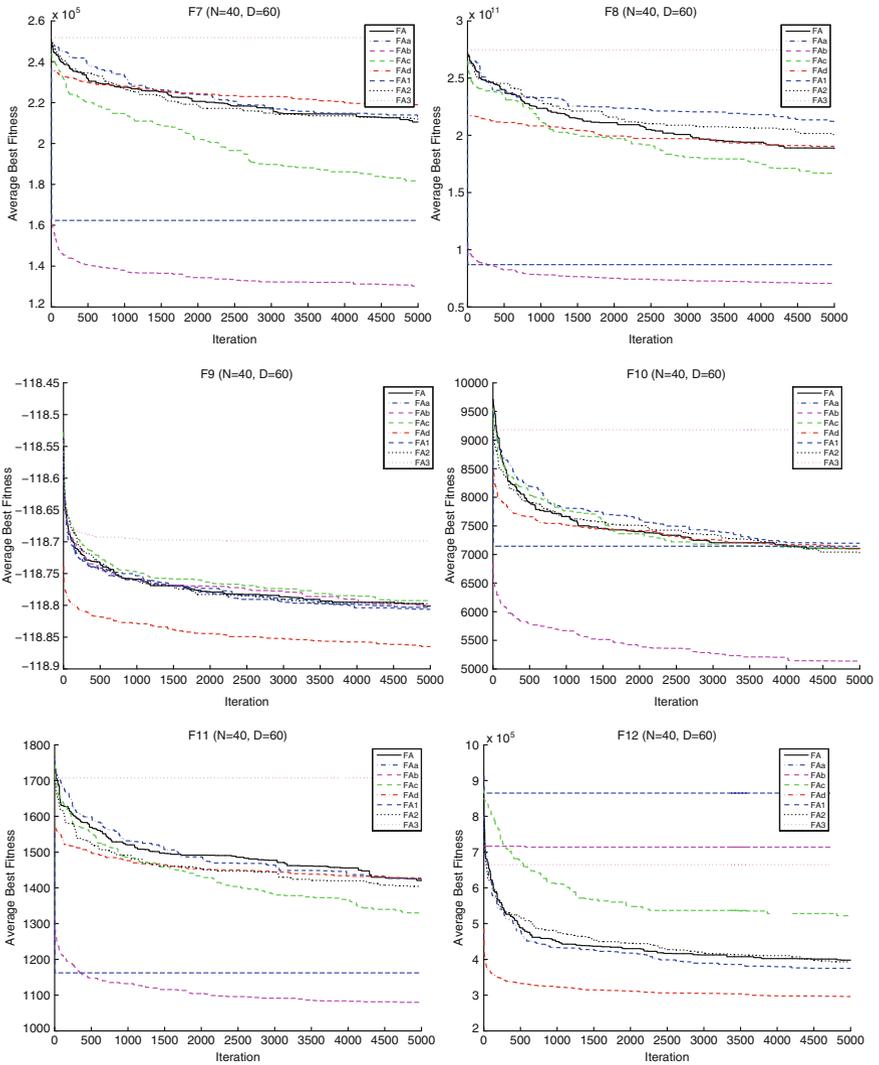


Fig. 2.7 Average best performance (averaged over 30 trials) of each algorithm variant on test problems F7–F12 (60D)

2.5.3 Analysis of Components in FA

Comparing the mean (of the best results found across each of the 30 trials) performance of FA with FA1, FA outperforms FA1 in 5 out of 12 cases (30D) and 4 out of 12 cases (60D). In all but one case, the difference in mean performance between the algorithms is significant. Hence, the conclusion drawn is that there is no compelling

evidence that the addition of a momentum mechanism has led to enhanced search performance. It is noted that the inclusion of a momentum mechanism in these experiments was motivated by general findings in the behavioural ecology literature [39] that organisms tend to display directional persistence rather than it being a distinct mechanism displayed by golden shiner fish [3].

Comparing the performance of FA with FA2, we note that FA outperforms FA2 in 7 out of 12 cases (30D) and 8 out of 12 cases (60D). In 9 cases (30D) and 1 case (60D) the difference is statistically significant. The conclusion drawn is that FA slightly outperforms FA2, but that the degree of outperformance becomes less (statistically speaking) as we move to the 60D case.

Comparing FA with FA3, FA outperforms FA3 in 11 out of 12 cases (30D) and 10 out of 12 cases (60D). In 11 cases (30D) and 8 cases (60D) the difference is statistically significant. The conclusion drawn is that FA generally outperforms FA3 and that, based on the results for FA2 and FA3, the inclusion of both IP and DP mechanisms (as distinct from only including one mechanism) produces a better quality search process.

We also compare the performance of FA2 and FA3, and find that FA2 outperforms FA3 in 7 out of 12 cases (30D) and 9 out of 12 cases (60D), indicating that a IP mechanism produces a better search performance than DP alone. This is not surprising as the DP mechanism is not driven by any feedback from the environment, and therefore, on its own is similar to a random search process. As would be expected, the standard deviation of the results produced by FA3 is generally higher than those produced by either FA1 or FA2.

Hence, the results suggest that while social information propagation can usefully spread information on good locations amongst the population of agents, it needs to be informed by information from the individual perception mechanism in order to strongly guide the search process. Combining the results, across the algorithmic variants we get a general performance ordering of $FA1 > FA > FA2 > FA3$.

2.5.4 Parameter Sensitivity Analysis

The detailed 'end of run' results from the FAa, FAb, FAc and FAd variant algorithms are shown in Tables 2.4, 2.5, 2.6 and 2.7. Initially, we compare the results of each algorithmic variant with the performance of the canonical algorithm FA.

We note that FA outperforms FAa in 8 out of 12 cases (30D) and 9 out of 12 cases (60D). In 9 cases (30D) and 4 cases (60D) these differences are statistically significant. This suggests that FA generally performs better than FAa, which is not unexpected given that FA has a wider perception radius.

Examining FA versus FAb, FA performs better in only 2 out of 12 cases (30D) and 1 out of 12 cases (60D). The differences in mean performance are statistically significant in 10 (30D) and 11 (60D) cases respectively. The strong performance of FAb arises as in this variant of the algorithm, the fish have a wider perception radius than they do in FA.

Comparing the results of FAa and FAb we note that FAa performs better on the majority of test problems. Combining the results from the above analyses, we can conclude that the choice of perception radius is a critical parameter for the algorithm.

The FAc variant employs 10 samples in each IP step. The canonical FA outperforms FAc in 12 out of 12 cases (30D) but in only 3 out of 12 cases (60D), with the differences in performance being significant in 8 (30D) and 10 (60D) cases respectively. It is interesting to note the switch in relative performance when the dimensionality of the test problems is increased.

In contrast to FAc, the variant FAd only undertakes a single sampling in each IP step. Comparing FA with FAd, FA performs better in 5 out of 12 cases (30D) and in 7 out of 12 cases (60D). The differences in performances are significant in 12 out of 12 cases (30D) and 6 out of 12 cases (60D).

Comparing FAc and FAd, it is not clearly evident that either outperforms the other, as the performance ranking between the two varies across the test problems. The conclusion is that the results from the FA algorithm are not clearly impacted by choice of number of IP samplings.

2.6 Conclusions

The propagation of information about the environment amongst a population via social communication has attracted increasing research interest in recent decades with the realisation that many animal species engage in subtle forms of information transfer which had previously escaped notice. More generally, we can consider this process as information flowing in a network of nodes or agents, wherein each agent receives inputs from their senses and from conspecifics, processes this information, and in turn through their resulting actions, subsequently influence actions of other agents.

In this study we draw inspiration from the schooling behaviour of ‘golden shiner’ fish which alter their movement in an effort to track shade and develop a novel optimization algorithm, the fish algorithm (FA). The FA can be considered as a swarm algorithm as the search process embeds bottom-up learning via information flow between agents (fish). We assess the utility of the algorithm on a series of test problems and undertake an analysis of the algorithm by examining the importance of its component elements for the search process. The results indicate that momentum or ‘directional persistence’ mechanism is not found to be particularly useful but that best results are obtained when using a mix of information from individual perception and social communication. While social communication can usefully spread information on good locations amongst the population of agents, it needs to be supplemented by information from the individual perception mechanism in order to strongly guide the search process.

The current study indicates several interesting areas for follow up research. Obviously the results from any study only extend to the test problems and specific parameter settings examined, and future work could seek to examine the utility

of the algorithm in additional problem domains. A factor which is not fully included in current work is that fish do not select shoal mates randomly but rather prefer to shoal with healthy fish, and fish which are similar in size and age to themselves. The algorithms developed in this chapter could be adapted to incorporate these issues more comprehensively.

At an even deeper level, the results of the study highlight the question as to what is the optimal balance between the use of individual perception and the propagation of social information in the population? In other words, what weight should be placed on each factor in order to optimise the search process. Further investigation of this issue has potential to assist in our understanding as to how best to tailor optimization algorithms for specific problem environments, and for deepening our understanding of the foraging strategies of various organisms.

Acknowledgments This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant Number 08/SRC/FM1389.

References

1. Amintoosi, M., Fathy, M., Mozayani, N., Rahmani, A.: A fish school clustering algorithm: applied to student sectioning problem. In: Proceedings of 2007 International Conference on Life System Modelling and Simulation (LSMS), Published as a Supplementary Volume to Dynamics of Continuous Discrete and Impulse Systems, Series B: Applications and Algorithms, vol. 2, pp. 696–699. Watam Press, Canada (2007)
2. Bastos Filho, C., de Lima Neto, F., Lins, A., Nascimento, A., Lima, M.: A novel search algorithm based on fish school behavior. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 2646–2651. IEEE Press, New York (2008)
3. Berdahl, A., Torney, C., Ioannou, C., Faria, J., Couzin, I.: Emergent sensing of complex environments by mobile animal groups. *Science* **339**, 574–576 (2013)
4. Bin, Z., Jianlin, M., Haiping, L.: A hybrid algorithm for sensing coverage problem in wireless sensor networks. In: Proceedings of IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, pp. 162–165. IEEE Press, Kunming (2011)
5. Bing, Z., Wen, D.: Scheduling arrival aircrafts on multi-runway based on improved artificial fish swarm algorithm. In: Proceedings of the 10th International Conference on Computational and Information Sciences (ICCIS '10), pp. 499–502. IEEE Press, New York (2010)
6. Bradbury, J., Vehrencamp, S.: Principles of Animal Communication, 2nd edn. Sinauer Associates, Sunderland, MA, USA (2011)
7. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, Oxford (1999)
8. Chong, C., Low, M., Sivakumar, A., Gay, K.: A bee colony optimization algorithm to job shop scheduling. In: Proceedings of the 2006 Winter Simulation Conference (WinterSim), pp. 1954–1961. IEEE Press, New Jersey (2006)
9. Dorigo, M.: Optimization, learning and natural algorithms. Ph.D. thesis, Politecnico di Milano (1992)
10. Dorigo, M., DiCaro, G.: Ant colony optimization: a new meta-heuristic. In: Proceedings of IEEE Congress on Evolutionary Computation (CEC), pp. 1470–1477. IEEE Press, Piscataway, NJ (1999)
11. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern.* **26**, 29–41 (1996)

12. Dorigo, M., Stützle, T.: *Ant Colony Optimization*. MIT Press, Cambridge (2004)
13. Engelbrecht, A.: *Fundamentals of Computational Swarm Intelligence*. Wiley, Chichester (2005)
14. Farzi, S.: Efficient job scheduling in grid computing with modified artificial fish swarm algorithm. *Int. J. Comput. Theory Eng.* **1**(1), 13–18 (2009)
15. Grunbaum, D., Viscido, S., Parrish, J.: Extracting interactive control algorithms from group dynamics of schooling fish. In: *Coop Control Lecture Notes in Control and Information Sciences (LNCIS 309)*, pp. 103–117. Springer (2004)
16. He, D., Qu, L., Guo, X.: Artificial fish-school algorithm for integer programming. In: *Proceedings of IEEE International Conference on Information Engineering and Computer Science (ICIECS)*, pp. 1–4. IEEE Press, New York (2009)
17. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948. IEEE Press, Piscataway, NJ (1995)
18. Kennedy, J., Eberhart, R., Shi, Y.: *Swarm Intelligence*. Morgan Kaufman, San Mateo (2001)
19. Li, X., Shao, Z., Qian, J.: An optimizing method based on autonomous animats: fish swarm algorithm. *Syst. Eng. Theory Pract.* **22**, 32–38 (2002) (in Chinese)
20. Liang, J.J., Suganthan, P.N., Deb, K.: Novel composition test functions for numerical global optimization. In: *Proceedings of IEEE Swarm Intelligence Symposium*, pp. 68–75. IEEE Press, Chicago (2005)
21. Liu, B.-Q., Sun G.-C.: Artificial fish swarm algorithm for traffic network design problem. *Comput. Eng.* **37**(8), 161–163 (2011) (in Chinese)
22. Kavanau, J.: Vertebrates that never sleep: implications for sleep's basic function. *Brain Res. Bull.* **46**(4), 269–279 (1998)
23. Nakrani, S., Tovey, C.: On honey bees and dynamic server allocation in internet hosting centres. *Adapt. Behav.* **12**, 223–240 (2004)
24. Neshat, M., Sepidnam, G., Sargolzaei, M., Najaran Toosi, A.: Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications. *Artif. Intell. Rev.* (2012). doi:[10.1007/s10462-012-9342-2](https://doi.org/10.1007/s10462-012-9342-2). Accessed 6 May 2012
25. Meng, F., Zhao, H., Zhao, Q., Ma, W., Cao, Y., Wang, L.: Artificial fish swarm-based energy efficient qos classification algorithm to next generation electric power communication networks. *Appl. Mech. Mater.* **392**, 857–861 (2013)
26. Miller, N., Garnier, S., Hartnett, A., Couzin, I.: Both information and social cohesion determine collective decisions in animal groups. *PNAS*, **110**(13), 5263–5268 (2013)
27. Parrish, J., Viscido, S., Grunbaum, D.: Self-organized fish schools: an examination of emergent properties. *Biol. Bull.* **202**, 296–305 (2002)
28. Passino, K.: Distributed optimization and control using only a germ of intelligence. In: *Proceedings of the 2000 IEEE International Symposium on Intelligent Control*, pp. 5–13. IEEE Press, Patras, Greece (2000)
29. Passino, K.: Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst. Mag.* **22**, 52–67 (2002)
30. Pham, D., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M.: The bees algorithm—a novel tool for complex optimization problems. In: *Proceedings of International Production Machines and Systems (IPROMS)*, pp. 454–459. Elsevier, Oxford (2006)
31. Reynolds, C.: Flocks, Herds and Schools, a distributed behavioral model. In: *Proceedings of the 14th annual conference on computer graphics and interactive techniques (SIGGRAPH)*, pp. 25–34. Anaheim, California (1987)
32. Slobodchikoff, C., Perla, B., Verdolin, J.: *Prairie Dogs: Communication and Community in an Animal Society*. Harvard University Press, Cambridge, Massachusetts (2009)
33. Stocker, S.: Models for tuna formation. *Math. Biosci.* **156**, 167–190 (1999)
34. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A., Tiwari, S.: Problem definitions and evaluation criteria for the CEC 2005 special session on Real-Parameter optimization. Nanyang Technological University, Technical report (2005)
35. Sumpter, D., Krause, J., James, R., Couzin, I., Ward, A.: Consensus decision making by fish. *Curr. Biol.* **18**, 1773–1777 (2008)

36. Tian, W., Liu, J.: An improved artificial fish swarm algorithm for multi robot task scheduling. In: Proceedings of the 2009 IEEE 5th International Conference on Natural Computation, pp. 127–130. IEEE Press, New York (2009)
37. Tian, W., Tian, Y., Ai, L., Liu, J.: A new optimization algorithm for fuzzy set design. In: Proceedings of the 2009 IEEE International Conference on Intelligent Human-Machine Systems and Cybernetics, pp. 431–435. IEEE Press, New York (2009)
38. Tsai, H.-C., Lin, Y.-H.: Modification of the fish swarm algorithm with particle swarm optimization formulation and communication behavior. *Appl. Soft Comput.* **11**, 5367–5374 (2011)
39. Viswanathan, G., da Luz, M., Raposo, E., Stanley, E.: *The Physics of Foraging: An Introduction to Random Searches and Biological Encounters*. Cambridge University Press, Cambridge (2011)
40. Wang, C.-R., Zhou, C.-L., Ma, J.-W.: An improved artificial fish-swarm algorithm and its application in feed-forward neural networks. In: Proceedings of the 2005 IEEE International Conference on Machine Learning and Cybernetics, vol. 5, pp. 2890–2894. IEEE Press (2005)
41. Yang, X.-S.: Engineering optimization via nature-inspired virtual bee algorithms. In: Mira, J., Álvarez, J. (eds.) *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach*, First International Work-Conference on the Interplay Between Natural and Artificial Computation (IWINAC 2005), pp. 317–323. Springer, Berlin (2005)
42. Yiyue, W., Hongmei, L., Hengyang, H.: Wireless sensor network deployment using an optimized artificial fish swarm algorithm. In: Proceedings of 2012 IEEE International Conference on Computer Science and Electronics Engineering, vol. 2, pp. 90–94. IEEE Press, Hangzhou (2012)
43. Zhang, K., Zhang, W., Dai, C.-Y., Zeng, J.-Z.: Artificial fish-swarm based coverage-enhancing algorithm for visible light sensor networks. *Optoelectron. Lett.* **6**(3), 229–231 (2010)
44. Zhou, Y., Liu, B.: Two novel swarm intelligence clustering analysis methods. In: Proceedings of the 5th International Conference on Natural Computation, pp. 497–501. IEEE Press, Tianjin (2009)



<http://www.springer.com/978-3-319-15915-7>

Propagation Phenomena in Real World Networks

Król, D.; Fay, D.; Gabryś, B. (Eds.)

2015, XVIII, 364 p. 97 illus., 32 illus. in color., Hardcover

ISBN: 978-3-319-15915-7