Distributed Perception Algorithm

Anthony Brabazon and Wei Cui

Natural Computing Research and Applications Group,
School of Business, University College Dublin, Ireland.
anthony.brabazon@ucd.ie, will.weicui@gmail.com

Abstract. In this paper we describe the *Distributed Perception Algorithm* (DPA) which is partly inspired by the schooling behaviour of 'golden shiner' fish (*Notemigonus crysoleucas*). These fish display a preference for shaded habitat and recent experimental work has shown that the fish use both individual and distributed perception in navigating their environment. We assess the contribution of each element of the DPA and also benchmark its results against those of canonical PSO.

1 Introduction

The last few decades have seen significant and growing interest in *biomimicry* or 'learning from the natural world', with many disciplines turning to natural phenomena for inspiration as to how to solve particular problems. Examples include the development of pharmaceutical products based on naturally occurring chemicals, and inspiration for engineering designs based on structures and materials found in nature. Another strand of 'learning from nature' concerns the development of computational algorithms whose design is inspired by underlying natural processes which implicitly embed computation [6]. Mechanisms of collective intelligence and their application as practical problem-solving tools, has attracted considerable research attention leading to the development of several families of swarm-inspired algorithms including, ant-colony optimisation [7–9], particle swarm optimisation [10, 13, 14], bacterial foraging [16, 17], honey bee algorithms [18] 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 sharing of information among a population of agents in which individuals only possess local information. In this paper we extend an initial examination [5] of the *distributed perception algorithm* (DPA) and assess its performance against particle swarm optimisation.

2 Background

A number of previous studies have previously employed a fish school metaphor to develop algorithms for optimisation and clustering ([1, 12, 21, 23] 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) [15].

A practical issue that arises in attempting to develop an algorithm based on the behaviour of fish schools is that we have surprising little hard data on the behavioural mechanisms which underlie their activity. At the level of the individual, agents respond to their own sensory inputs and to their physiological / cognitive states [11]. 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. 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 paper, we draw inspiration from certain behaviours of the fish species 'golden shiners'. This is a fresh water fish which is native to North America, typically growing to about 4 to 5 inches in length. The species is strongly gregarious. Members of the species form shoals of up to about 250 individuals [19]. A recent study [4] investigated one aspect of the behaviour of golden shiners, namely their marked preference for shaded habitat. In order to investigate the mechanism underlying their observed collective response to light gradients, fish were tracked individually to obtain trajectories. The study examined the degree to which the motion of individuals is explained by individual perception (this would produce movement in the steepest direction of light gradient as seen by the individual fish) and social influences based on distributed perception (this would produce movement based on the position and movement of conspecifics). The study indicated that the relative importance of each is context dependent. For example, when the magnitude of the social vector was high (all conspecifics moving in similar direction) the social influence was dominant.

3 Distributed Perception Algorithm

An important question that underlies the design of foraging strategies, or the design of optimisation algorithms, is what is the most effective way of searching for objects whose location on a landscape is not known a priori? In foraging, the search could be guided by external cues, either via cognitive skills (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 [22]. More generally, it is noted that this framework can encompass most foraging-inspired algorithms, as virtually all of them embed personal perception / learning, social influence and a stochastic element.

In each iteration of the proposed DPA, a fish is displaced from its previous position through the application of a velocity vector:

$$p_{i,t} = p_{i,t-1} + v_{i,t} \tag{1}$$

where $p_{i,t}$ is the position of the i^{th} fish at current iteration, $p_{i,t-1}$ is the position of the i^{th} fish at previous iteration, 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:

$$v_{i,t} = v_{i,t-1} + DP_{i,t} + IP_{i,t}\ , \tag{2}$$

or more generally

$$v_{i,t} = w_1 v_{i,t-1} + w_2 DP_{i,t} + w_3 IP_{i,t} \ . \tag{3}$$

The difference between the two update equations is that weight coefficients are included in Eq. 3. In all the experiments of this study, Eq. 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 p_{best} and g_{best} in PSO. The next subsection explains the operation of the two perception mechanisms.

3.1 Prior Period Velocity

The inclusion of a prior period velocity can be considered as a proxy for momentum or inertia. The inclusion of this term is motivated by empirical evidence from the movement ecology literature which indicates that organisms do not follow uncorrelated random walks but rather move with a 'directional persistence' [22].

3.2 Distributed Perception Influence

The distributed perception influence for the i^{th} fish is determined by the following:

$$DP_i = \frac{\sum_{j=1}^{N_i^{DP}} (p_j - p_i)}{N_i^{DP}} \ , \qquad j \neq i \tag{4}$$

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 $0 <\mid p_j - p_i \mid \leq r_{DP}$, 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.

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 the function as below:

$$IP_i = \frac{\sum_{j=1}^{N_i^{IP}} (s_j - p_i) * fit_j}{\sum_{j=1}^{N_i^{IP}} fit_j} \ , \qquad j \neq i \tag{5}$$

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 ($0 <\mid s_j - p_i \mid \leq r_{IP}$), and fit_j is the fitness value of the j^{th} sample.

4 Results

In this section we describe the experiments undertaken and present the results from these experiments. Four standard benchmark problems (Table 1) were used to test the developed algorithms. Two of these functions namely, DeJong and Rosenbrock, represent unimodal problems; and the other two, Griewank and Rastrigin, are more complex functions with multiple local optima. The aim in all the experiments is to find the vector of values which minimise the value of the test functions.

Table 1. Optimisation Problems

DeJong	$F(x) = \sum_{i=1}^{n} x_i^2$	$[-5.12\ 5.12]^n$	0.0^n
Griewank	$F(x) = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}})$	$[-600\ 600]^n$	0.0^n
Rastrigin	$F(x) = 10n + \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i)]$	$[-5.12\ 5.12]^n$	0.0^n
Rosenbrock	$F(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	$[-30\ 30]^n$	1.0^n

In our experiments we assess the performance of the DPA on the four benchmark problems and also investigate the importance of the three components in the DPA algorithm, namely momentum, DP and IP using the DeJong function. The aim is to examine whether these components play a significant role in determining the DPA's performance. Three algorithms are developed which switch off in turn the momentum, DP and IP influences, denoted as DPAa1, DPAa2 and DPAa3 respectively. The performance of the three variants are compared with that of the standard DPA algorithm which has all three components (denoted as DPAa).

The second set of experiments examines the sensitivity of the DPA to changes in two of its parameters, namely the radius of perception (r_{DP}, r_{IP}) and the number of samples used in the simulated individual perception component (s). The chosen values of these parameters are shown in Table 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 DAPa algorithm, the values of r_{DP} and r_{IP} were chosen after initial experimentation as $\frac{R}{1.5\sqrt[D]{N}}$ and $\frac{R}{1.8\sqrt[D]{N}}$ so that in most cases each fish has neighbouring fish within the radius r_{DP}. In order to undertake some sensitivity analysis, two variants of the DPAa algorithm are developed. In the DPAb algorithm, the values of r_{DP} and r_{IP} are set to be twice as large as those in the DPAa algorithm. In the DPAc algorithm, the value of s is increased to 10 (as against 5 in the DPAa algorithm).

Finally, the results from the DPA are compared against those of canonical Particle Swarm Optimisation (PSO). In order to allow a reasonably fair comparison, we control for the number of fitness function evaluations. The canonical PSO algorithm is run for five times as many iterations as DPA, as each canonical version of the DPA undertakes five fitness samplings in the IP mechanism.

Table 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
DPAa	$\frac{R}{1.5\sqrt[D]{N}}$	$\frac{R}{1.8\sqrt[D]{N}}$	5	$v_{i,t} = v_{i,t-1} + DP_{i,t} + IP_{i,t}$
DPAa1	$\frac{R}{1.5\sqrt[D]{N}}$	$\frac{R}{1.8\sqrt[D]{N}}$	5	$v_{i,t} = 0 + DP_{i,t} + IP_{i,t}$
DPAa2	$\frac{R}{1.5\sqrt[D]{N}}$	$\frac{R}{1.8\sqrt[D]{N}}$	5	$v_{i,t} = v_{i,t-1} + 0 + IP_{i,t}$
DPAa3	$\frac{R}{1.5\sqrt[D]{N}}$	$\frac{R}{1.8\sqrt[D]{N}}$	5	$v_{i,t} = v_{i,t-1} + DP_{i,t} + 0$
DPAb	$\frac{R}{3\sqrt[D]{N}}$	$\frac{R}{3.6\sqrt[D]{N}}$	5	$v_{i,t} = v_{i,t-1} + DP_{i,t} + IP_{i,t}$
DPAc	$\frac{R}{1.5\sqrt[D]{N}}$	$\frac{R}{1.8\sqrt[D]{N}}$	10	$v_{i,t} = v_{i,t-1} + DP_{i,t} + IP_{i,t}$

Note: R is the radius of the search space, D is the dimension of the test problem and N is the number of fish.

4.1 Hypotheses and Parameter Settings

In all the experiments, we undertake thirty runs of each algorithm and average the results obtained over these runs. In order to assess the relative performance of each algorithm variant we examine the statistical significance of differences in performance at a conservative 99% level using a t-test.

The first set of hypotheses concern the testing of the importance of each component of the DPA. The null hypothesis is that the algorithm with a component turned off performs better than the canonical DPA (DPAa). Therefore three hypotheses are tested as follows.

- H_{a1}: the DPAa1 algorithm outperforms the DPAa algorithm;
- H_{a2}: the DPAa2 algorithm outperforms the DPAa algorithm;
- H_{a3}: the DPAa3 algorithm outperforms the DPAa algorithm;

The next set of hypotheses concern the analysis of the two variants with different parameter settings (DPAb and DPAc) of the canonical algorithm (DPAa).

- H_{ba}: the DPAb algorithm outperforms the DPAa algorithm;
- H_{ca}: the DPAc algorithm outperforms the DPAa algorithm;

The final set of hypotheses concern the analysis of the performance of the three versions of the canonical algorithm with PSO.

- H_{a0}: the PSO algorithm outperforms the DPAa algorithm;
- H_{b0}: the PSO algorithm outperforms the DPAb algorithm;
- H_{c0}: the PSO algorithm outperforms the DPAc algorithm.

In all experiments, 30 fish (in DPAs), or in the case of PSO 30 particles, are used. The results are described in the following three sections.

Table 3. Results of Component Analysis

		DeJong (20D)	DeJong (40D)	DeJong (60D)
DPAa	**Best**	13.8490	39.8309	82.9444
	Mean	16.7810	53.3594	99.3693
	Std Dev	1.6621	4.4814	5.4045
DPAa1	**Best**	101.3639	146.8877	173.8090
	Mean	108.8814	154.1140	190.1412
	Std Dev	3.3446	4.5209	7.5146
	H_{a1}	0.00	0.00	0.00
DPAa2	**Best**	76.9869	211.8659	384.0815
	Mean	88.1540	261.8146	459.8905
	Std Dev	6.2977	17.8869	21.6424
	H_{a2}	0.00	0.00	0.00
DPAa3	**Best**	104.0716	346.6357	522.4203
	Mean	181.4443	409.8456	636.0020
	Std Dev	29.0740	38.0923	58.6390
	H_{a3}	0.00	0.00	0.00

4.2 Analysis of Components in DPA

The developed algorithms, DPAa, DPAa1, DPAa2 and DPAa3, were tested on the DeJong problem with 20, 40, and 60 dimensions respectively. Fig. 1 compares the average fitness of the four algorithms for the three tested problems. Table 3 shows the best fitness value obtained from all 30 runs ('Best'), the average of the best fitness ('Mean') and its standard deviation over all 30 runs. The results show that the standard DPA algorithm (DPAa) significantly outperforms the other three algorithms, which indicates that none of the three components, momentum, DP and IP, are sufficient on their own to produce a good search process. It is also observed that DPAa3 (which has individual perception 'switched off') performs the worst. This is not surprising as the IP component is fitness-guided.

We also carry out a statistical significance test of the differences on performance between the DPAa algorithm and the other three algorithms, namely DPAa1, DPAa2 and DPAa3 and the p-values are shown in Table 3. The results indicate that the DPAa algorithm outperforms the DPAa1, DPAa2 and DPAa3 algorithms on all tested problems at a significance level of 0.99.

4.3 Parameter Sensitivity Analysis

The results of the DPAa, DPAb and DPAc algorithms are shown in Table 4 and Fig. 1. As can be seen, the DPAa and DPAc algorithms perform significantly better than the DPAb algorithm, which indicates that the radii of DP and IP are a critical factor in determining the performance of DPA algorithm. A larger radius means that the fish can sample more broadly in the IP component and can be influenced by more fish in the DP component of the algorithm.

Comparing the DPAa and DPAc variants of the algorithm (these focus on the sensitivity of the results to the number of samples in the IP mechanism), the results in Fig. 1 show that the DPAc tends to do slightly better than DPAa but this difference is not generally statistically significant. This indicates that the results obtained are not crucially dependent on the number of samples used in the IP mechanism.

Table 4. Results of Algorithm Comparison

		DeJong (60D)	Griewank (60D)	Rastrigin (60D)	Rosenbrock (60D)
PSO	**Best**	229.6262	705.1046	706.8452	1.8195E+8
	Mean	313.7544	1059.4572	809.0102	4.2095E+8
	Std Dev	41.3335	205.1225	57.0805	1.0482E+8
DPAa	**Best**	82.9444	314.6526	499.8868	0.4054E+8
	Mean	99.3693	340.4509	535.9554	0.5268E+8
	Std Dev	5.4045	14.8251	15.9696	0.0591E+8
	H_{a0}	0.00	0.00	0.00	
DPAb	**Best**	454.0024	1459.6416	872.9574	8.2966E+8
	Mean	488.4168	1670.7541	947.8529	9.1680E+8
	Std Dev	17.5184	71.9513	24.9544	0.3978E+8
	H_{b0}	1.00	1.00	1.00	
	H_{ba}	0.00	0.00	0.00	
DPAc	**Best**	66.9831	251.4181	492.2972	0.2669E+8
	Mean	88.2610	315.1278	530.1059	0.4270E+8
	Std Dev	8.2310	24.7452	13.4010	0.0689E+8
	H_{c0}	0.00	0.00	0.00	
	H_{ca}	1.00	1.00	1.00	

4.4 Comparisons with Canonical PSO

The comparisons between the DPA algorithms and the PSO algorithm are shown in Table 4 and Fig. 1. Two of the DPA algorithm variants, DPAa and DPAc, are seen to outperform the PSO algorithm across all benchmarks. The standard deviations of all the DPA algorithms are smaller than for the PSO algorithm.

5 Conclusions

In this paper we describe the distributed perception algorithm which is inspired by schooling behaviour of ‘golden shiner’ fish. We assess the utility of the algorithm on

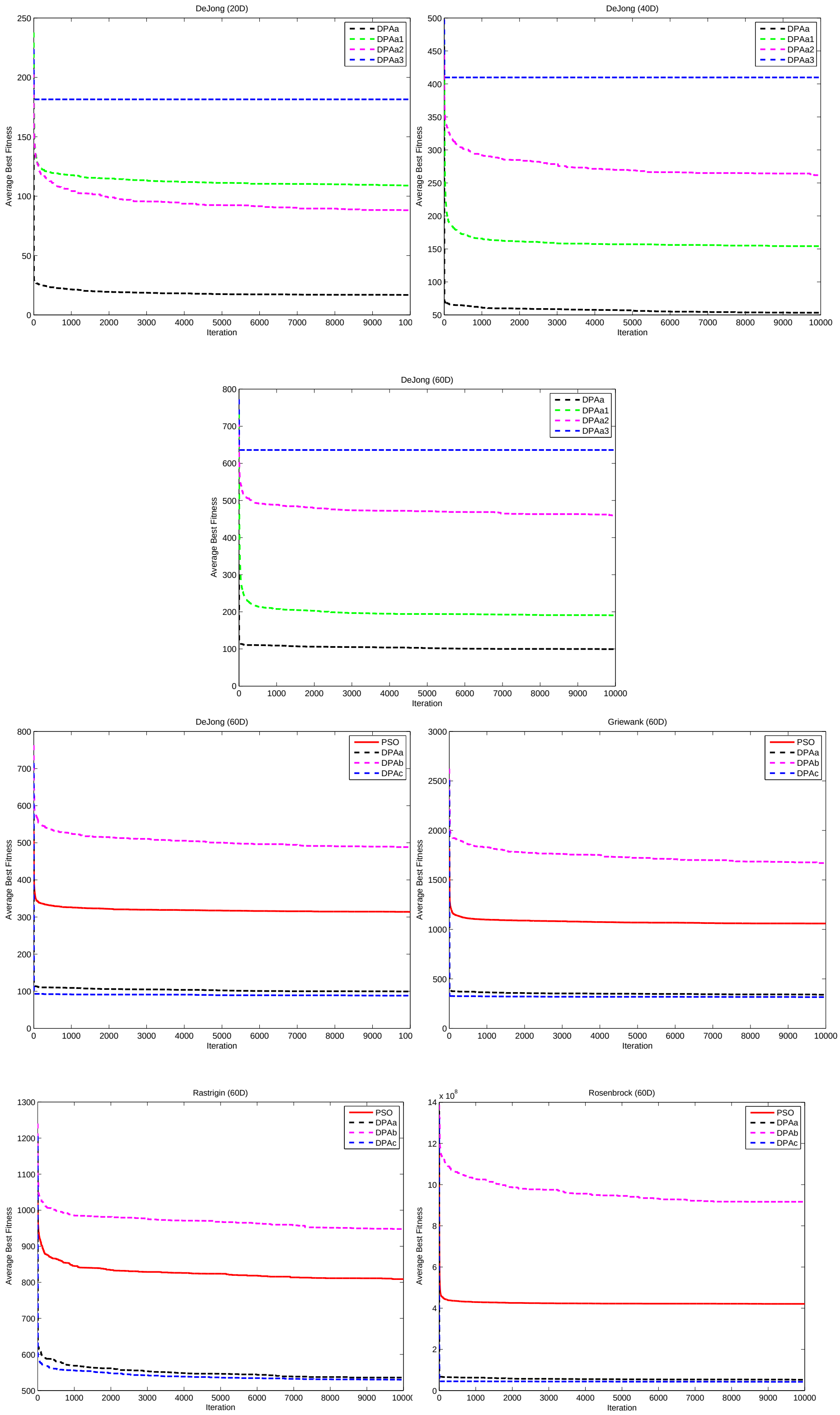

Fig. 1. Top three figures analyse components in DPA using DeJong 20,40 and 60D cases (x-axis shows the iteration number). The bottom four figures compare DPA variants vs PSO on four 60D problems

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 obtained are benchmarked against those from particle swarm optimisation (PSO). The results indicate that the algorithm is competitive against canonical PSO and support a claim that algorithms employing fish-school behaviour mechanisms can be a useful addition to an optimisation toolkit. The current study indicates several interesting areas for follow up research. Obviously the results from any study only extend to the problems examined and future work is required to examine the utility of the algorithm. It is also noted that other fish school algorithms have been developed using search mechanisms inspired by various fish behaviours and as this area of research matures, it would be useful to integrate these into a broader, general, framework.

Another interesting avenue would be to investigate alternative ways of modelling the distributed perception (DP) mechanism. In this study, following [4], we assume that this sensory mechanism has a fixed metric range, in other words, a fish 'interacts' with all its neighbours within a defined radius. Alternative assumptions as to the nature of interaction range can be made including [20], a fixed number of nearest neighbours (topological range) or a shell of near neighbours (Voronoi range). In [20], a novel approach is adopted whereby each fish is assigned a 'visual field' and only neighbours within this field impact on the social information processed by that fish. An interesting finding of this work is that there is lower redundancy of information (transitivity) in visually-defined networks than in metric or topological networks. This could be a useful characteristic in the context of designing an optimisation algorithm, particularly for application in a dynamic environment.

Acknowledgement 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), Watam Press (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), IEEE Press, pp 2646–2651 (2008)
3. Bastos Filho, C., de Lima Neto, F., Sousa, M., Pontes, M., Madeiro, S.: On the Influence of the Swimming Operators in the Fish School Search Algorithm. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC), IEEE Press, pp 5012–5017 (2009)
4. Berdahl, A., Torney, C., Ioannou, C., Faria, J., Couzin, I.: Emergent Sensing of Complex Environments by Mobile Animal Groups. Science 339, 574– 576 (2013)
5. Brabazon, A., Cui, W., O'Neill, M.: Information Propagation in a Social Network: The Case of The Fish Algorithm. In: Krol D, Fay D, Gabrys B (eds) Propagation Phenomena in Real World Networks, pp 27–51 Springer (2015)
6. Brabazon, A., O'Neill, M., McGarraghy, S.: Natural Computing Algorithms. Springer (2015)

7. Dorigo, M.: Optimization, Learning and Natural Algorithms, PhD Thesis, Politecnico di Milano (1992)
8. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, And Cybernetics - Part B: Cybernetics, 26, 29–41 (1996)
9. Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press, Cambridge, Massachusetts (2004)
10. Engelbrecht, A.: Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, Chichester, England (2005)
11. Grunbaum, D., Viscido, S., Parrish, J.: Extracting interative control algorithms from group dynamics of schooling fish. Coop Control Lecture Notes in Control and Information Sciences (LNCIS 309), pp 103-117, Springer (2004)
12. 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), IEEE Press, pp 1–4 (2009)
13. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, IEEE Press, pp 1942–1948 (1995)
14. Kennedy, J., Eberhart, R., Shi, Y.: Swarm Intelligence. Morgan Kaufman, San Mateo, California (2001)
15. Li, X., Shao, Z., Qian, J.: An optimizing method based on autonomous animats: fish swarm algorithm. Systems Engineering Theory and Practice, 22, 32–38 (2002)
16. Passino, K.: Distributed Optimization and Control Using Only a Germ of Intelligence. In: Proceedings of the 2000 IEEE International Symposium on Intelligent Control, IEEE Press, pp 5–13 (2000)
17. Passino, K.: Biomimicry of Bacterial Foraging for Distributed Optimization and Control. IEEE Control Systems Magazine 22, 52-67 (2002)
18. Pham, D., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M.: The Bees Algorithm - A novel tool for complex optimisation problems. In: Proceedings of International Production Machines and Systems (IPROMS), pp 454–459 (2006)
19. Reebs, S.: Can a minority of informed leaders determine the foraging movements of a fish shoal? Animal Behaviour 59, 403–409 (2000)
20. Strandburg-Peshkin, A., Twomey, C., Bode, N., Kao, A., Katz, Y., Ioannou, C., Rosenthal, S., Torney, C., Wu, H., Levin, S., Couzin, I.: Visual sensory networks and effective information transfer in animal groups. Current Biology 23(17), R709–R711 (2013)
21. Tsai, H-C., Lin, Y-H.: Modification of the fish swarm algorithm with particle swarm optimization formulation and communication behavior. Applied Soft Computing 11, 5367–5374 (2011)
22. Viswanathan, G., da Luz, M., Raposo, E., Stanley, E.: The Physics of Foraging: An Introduction to Random Searches and Biological Encounters. Cambridge University Press (2011)
23. Zhou, Y., Liu, B.: Two Novel Swarm Intelligence Clustering Analysis Methods. In: Proceedings of the 5th International Conference on Natural Computation, IEEE Press, pp 497–501 (2009)