# Feeding the Fish – Weight Update Strategies for the Fish School Search Algorithm

Andreas Janecek

北京大學 PEKING UNIVERSITY

andreas.janecek@univie.ac.at

International Conference on Swarm Intelligence (ICSI)

Chongqing, China - Jun 14, 2011

# Outline

1. Basic Fish School Search
   - Motivation
   - Basic Structure
   - Operators
2. Update Strategies
   - Weight Update Strategies
   - Non-linear Step Size Decrease Strategy
   - Combined Strategy
3. Evaluation
   - Fitness per Iteration
   - Final Results

# Overview

## What is Fish School Search?

- FSS is a recently developed swarm intelligence algorithm based on the social behavior of schools of fish

## Connection to Fish Swarms in Biology

- By living in swarms, the fish improve survivability of the whole group due to mutual protection against enemies
- The fish perform collective tasks in order to achieve synergy (e.g. finding locations with lots of food)
- Comparable to real fish that swim in the aquarium in order to find food, the artificial fish search (***swim***) the search space (***aquarium***) for the best solutions (locations with most ***food***)

# Overview

## What is Fish School Search?

- FSS is a recently developed swarm intelligence algorithm based on the social behavior of schools of fish

## Connection to Fish Swarms in Biology

- By living in swarms, the fish improve survivability of the whole group due to mutual protection against enemies
- The fish perform collective tasks in order to achieve synergy (e.g. finding locations with lots of food)
- Comparable to real fish that swim in the aquarium in order to find food, the artificial fish search (***swim***) the search space (***aquarium***) for the best solutions (locations with most ***food***)

# Overview

## Inventors

- Carmelo J. A. Bastos Filho and
- Fernando Buarque de Lima Neto

Computational Intelligence Research Group (CIRG) at University of Pernambuco, Recife-PE, Brazil

## First Publications

- BASTOS-FILHO; LIMA NETO, *et al*. ***Fish School Search: An Overview***. In: CHIONG, Raymond (Ed.). Nature-Inspired Algorithms for Optimisation. Series: Studies in Computational Intelligence, Vol. 193.. pp. 261-277. Berlin: Springer, 2009
- BASTOS-FILHO; LIMA NETO, *et al*. ***On the Influence of the Swimming Operators in the Fish School Search Algorithm.*** In: IEEE International Conference on Systems, Man, and Cybernetics - SMC2009, 2009, San Antonio, USA.

# Overview

## Inventors

- Carmelo J. A. Bastos Filho and
- Fernando Buarque de Lima Neto

Computational Intelligence Research Group (CIRG) at University of Pernambuco, Recife-PE, Brazil

## First Publications

- BASTOS-FILHO; LIMA NETO, *et al*. ***Fish School Search: An Overview***. In: CHIONG, Raymond (Ed.). Nature-Inspired Algorithms for Optimisation. Series: Studies in Computational Intelligence, Vol. 193.. pp. 261-277. Berlin: Springer, 2009
- BASTOS-FILHO; LIMA NETO, *et al*. ***On the Influence of the Swimming Operators in the Fish School Search Algorithm.*** In: IEEE International Conference on Systems, Man, and Cybernetics - SMC2009, 2009, San Antonio, USA.

# Properties of FSS

Autonomy – no global information needs to be stored

Autonomy – only local computations

Low communications – minimum centralized control

Scalability / Parallelism

Non-monotonicity – distinct diversity mechanisms

Simple computations

# Outline

Basic Structure

# FSS Pseudo Code

```
initialize randomly all fish;
while stop criterion is not met do
    for each fish do
        individual movement
        + evaluate fitness function;
        feeding operator;
    end
    for each fish do
        instinctive movement;
    end
    Calculate barycentre;
    for each fish do
        volitive movement;
        evaluate fitness function;
    end
    update step_ind
end
```

# FSS Pseudo Code – Operators

```
initialize randomly all fish;
while stop criterion is not met do
    for each fish do
        individual movement
        + evaluate fitness function;
        feeding operator;
    end
    for each fish do
        instinctive movement;
    end
    Calculate barycentre;
    for each fish do
        volitive movement;
        evaluate fitness function;
    end
    update step_ind
end
```

**'Feeding Operator':**

Updates the fish weight according to successfulness of the current movement

# FSS Pseudo Code – Operators

```
initialize randomly all fish;
while stop criterion is not met do
    for each fish do
        individual movement
        + evaluate fitness function;
        feeding operator;
    end
    for each fish do
        instinctive movement;
    end
    Calculate barycentre;
    for each fish do
        volitive movement;
        evaluate fitness function;
    end
    update step_ind
end
```

**'Feeding Operator':**

Updates the fish weight according to successfulness of the current movement

**'Swimming Operators':**

Move the fish according to feeding operator

1. Individual movement
2. Collective *instinctive* movem.
3. Collective *volitive* movement

# Variables

- Population size (size of fish school): pop ($1 \leq i \leq pop$)
- Problem dimension: dim ($1 \leq j \leq dim$)
- Time (i.e. iteration): $t$
- Weight of fish $i$: $w_i$
- Position of fish $i$: $\vec{x}_i$
- Fitness of fish $i$: $f(\vec{x}_i)$

# Outline

Operators

▸ Jump directly to new update strategies

# FSS Pseudo Code

```
initialize randomly all fish;
while stop criterion is not met do
    for each fish do
        individual movement
        + evaluate fitness function;
        feeding operator;
    end
    for each fish do
        instinctive movement;
    end
    Calculate barycentre;
    for each fish do
        volitive movement;
        evaluate fitness function;
    end
    update step_ind
end
```

# Operators (1) – Individual Movement

- In each iteration, each fish randomly chooses a new position
- This position is determined by adding to each dimension $j$ of the current position $\vec{x}$ a random number multiplied by a predetermined step ($step_{ind}$)

$$n_j(t) = x_j(t) + randu(-1,1) * step_{ind}$$

($randu(-1,1)$ is a random number from a uniform distribution in interval $[-1,1]$)

- The parameter $step_{ind}$ decreases linearly during the iterations

$$step_{ind}(t+1) = step_{ind}(t) - \frac{step_{ind\,initial} - step_{ind\,final}}{number\ of\ iterations}$$

# Operators (1) – Individual Movement

- In each iteration, each fish randomly chooses a new position
- This position is determined by adding to each dimension $j$ of the current position $\vec{x}$ a random number multiplied by a predetermined step ($step_{ind}$)

$$n_j(t) = x_j(t) + randu(-1,1) * step_{ind}$$

($randu(-1,1)$ is a random number from a uniform distribution in interval $[-1,1]$)

- The parameter $step_{ind}$ decreases linearly during the iterations

$$step_{ind}(t+1) = step_{ind}(t) - \frac{step_{ind\ initial} - step_{ind\ final}}{number\ of\ iterations}$$

# Operators (1) – Individual Movement cont'd

Remember (1) from before: $n_j(t) = x_j(t) + randu(-1,1) * step_{ind}$

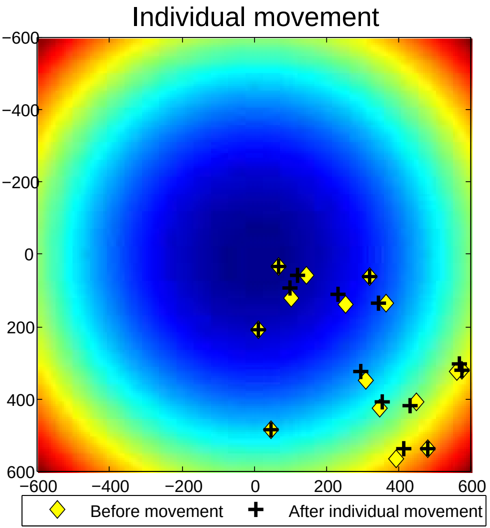
- The movement *only* occurs if the new position $\vec{n}$ has a better fitness than the current position $\vec{x}$, and if $\vec{n}$ lies within the aquarium boundaries
- Fitness difference ($\Delta f$) and displacement ($\Delta\vec{x}$) are evaluated according to

$$\Delta f = f(\vec{n}) - f(\vec{x})$$

$$\Delta\vec{x} = \vec{n} - \vec{x}$$

- If no individual movement occurs $\Delta f = 0$ and $\Delta\vec{x} = \vec{0}$

# Operators (1) – Individual Movement cont'd



# FSS Pseudo Code

```
initialize randomly all fish;
while stop criterion is not met do
    for each fish do
        individual movement
        + evaluate fitness function;
        feeding operator;
    end
    for each fish do
        instinctive movement;
    end
    Calculate barycentre;
    for each fish do
        volitive movement;
        evaluate fitness function;
    end
    update step_ind
end
```

# Operators (2) – Feeding

- Fish can increase their weight depending on the success of the individual movement according to

$$w_i(t+1) = w_i(t) + \frac{\Delta f(i)}{max(\Delta f)}$$

  - $w_i(t)$ is the weight of fish $i$
  - $\Delta f(i)$ is the difference of the fitness at current and new location
  - max($\Delta f$) is the maximum $\Delta f$ of all fish

- An additional parameter $w_{scale}$ limits the weight of a fish ($1 <= w_i <= w_{scale}$)

# Operators (2) – Feeding

- Fish can increase their weight depending on the success of the individual movement according to

$$w_i(t+1) = w_i(t) + \frac{\Delta f(i)}{max(\Delta f)}$$

  - $w_i(t)$ is the weight of fish $i$
  - $\Delta f(i)$ is the difference of the fitness at current and new location
  - max($\Delta f$) is the maximum $\Delta f$ of all fish

- An additional parameter $w_{scale}$ limits the weight of a fish ($1 <= w_i <= w_{scale}$)

# FSS Pseudo Code

```
initialize randomly all fish;
while stop criterion is not met do
    for each fish do
        individual movement
        + evaluate fitness function;
        feeding operator;
    end
    for each fish do
        instinctive movement;
    end
    Calculate barycentre;
    for each fish do
        volitive movement;
        evaluate fitness function;
    end
    update step_ind
end
```

# Operators (3) – Collective Instinctive Movement

- At first, a ***weighted average of individual movements*** $\vec{m}(t)$ based on success of individual movement of all fish is computed
- All fish that successfully performed individual movements ($\Delta\vec{x} \mathrel{!=} \vec{0}$) influence resulting direction of the school movement
- The resulting direction $\vec{m}(t)$ is evaluated by

$$\vec{m}(t) = \frac{\sum_{i=1}^{N} \Delta\vec{x}_i \Delta f_i}{\sum_{i=1}^{N} \Delta f_i}$$

- Finally, all fish update their positions according to $\vec{m}(t)$

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{m}(t)$$

# Operators (3) – Collective Instinctive Movement

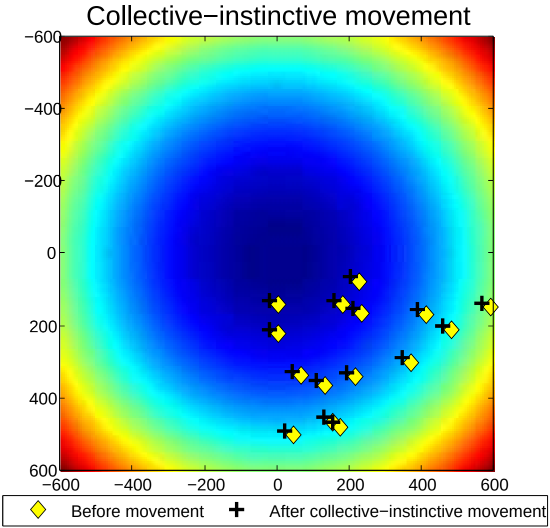
- At first, a ***weighted average of individual movements*** $\vec{m}(t)$ based on success of individual movement of all fish is computed
- All fish that successfully performed individual movements ($\Delta\vec{x} \mathrel{!=} \vec{0}$) influence resulting direction of the school movement
- The resulting direction $\vec{m}(t)$ is evaluated by

$$\vec{m}(t) = \frac{\sum_{i=1}^{N} \Delta\vec{x}_i \Delta f_i}{\sum_{i=1}^{N} \Delta f_i}$$

- Finally, all fish update their positions according to $\vec{m}(t)$

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{m}(t)$$

# Operators (3) – Collective Instinctive Movement cont'd



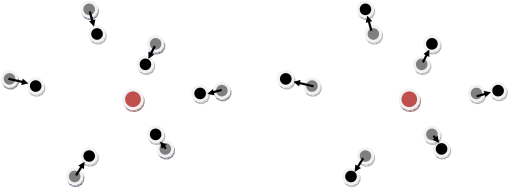
# FSS Pseudo Code

```
initialize randomly all fish;
while stop criterion is not met do
    for each fish do
        individual movement
        + evaluate fitness function;
        feeding operator;
    end
    for each fish do
        instinctive movement;
    end
    Calculate barycentre;
    for each fish do
        volitive movement;
        evaluate fitness function;
    end
    update step_ind
end
```

# Operators (4) – Collective Volitive Movement cont'd

Depending on the overall success rate of the whole school of fish, this movement is either a ...

- contraction of the swarm towards the barycenter of all fish, or
- dilation of the swarm away from the barycenter



Contraction Dilation

# Operators (4) – Collective Volitive Movement cont'd

- If the overall weight **in**creased after the individual movement step, the radius of the fish school is contracted in order to increase the exploitation ability
- If the overall weight **de**creased after the individual movement step, the radius of the fish school is dilated in order to cover a bigger area of the search space

**Barycenter (center of mass / gravity)**

Can be calculated based on the location $\vec{x}_i$ and weight $w_i$ of each fish

$$\vec{b}(t) = \frac{\sum_{i=1}^{N} \vec{x}_i w_i(t)}{\sum_{i=1}^{N} w_i(t)}$$

# Operators (4) – Collective Volitive Movement cont'd

- If the overall weight **in**creased after the individual movement step, the radius of the fish school is contracted in order to increase the exploitation ability
- If the overall weight **de**creased after the individual movement step, the radius of the fish school is dilated in order to cover a bigger area of the search space

## Barycenter (center of mass / gravity)

Can be calculated based on the location $\vec{x}_i$ and weight $w_i$ of each fish

$$\vec{b}(t) = \frac{\sum_{i=1}^{N} \vec{x}_i w_i(t)}{\sum_{i=1}^{N} w_i(t)}$$

# Operators (4) – Collective Volitive Movement cont'd

When the total weight of the school *increased* in the current iteration, all fish must update their location according to

$$\vec{x}(t+1) = \vec{x}(t) \; - \; step_{vol} randu(0,1) \frac{(\vec{x}(t) - \vec{b}(t))}{distance(\vec{x}(t), \vec{b}(t))}$$

When the total weight *decreased* in the current iteration the update is

$$\vec{x}(t+1) = \vec{x}(t) \; + \; step_{vol} randu(0,1) \frac{(\vec{x}(t) - \vec{b}(t))}{distance(\vec{x}(t), \vec{b}(t))}$$

- *distance*() is a function which returns the Euclidean distance between $\vec{x}$ and $\vec{b}$
- $step_{vol}$ is a predetermined step used to control the displacement from/to the barycenter ($step_{vol} = 2 * step_{ind}$ is a good choice)

# Operators (4) – Collective Volitive Movement cont'd

When the total weight of the school *increased* in the current iteration, all fish must update their location according to

$$\vec{x}(t+1) = \vec{x}(t) \; - \; step_{vol} randu(0,1) \frac{(\vec{x}(t) - \vec{b}(t))}{distance(\vec{x}(t), \vec{b}(t))}$$

When the total weight *decreased* in the current iteration the update is

$$\vec{x}(t+1) = \vec{x}(t) \; + \; step_{vol} randu(0,1) \frac{(\vec{x}(t) - \vec{b}(t))}{distance(\vec{x}(t), \vec{b}(t))}$$

- *distance*() is a function which returns the Euclidean distance between $\vec{x}$ and $\vec{b}$
- $step_{vol}$ is a predetermined step used to control the displacement from/to the barycenter ($step_{vol} = 2 * step_{ind}$ is a good choice)

# Operators (4) – Collective Volitive Movement cont'd

When the total weight of the school *increased* in the current iteration, all fish must update their location according to

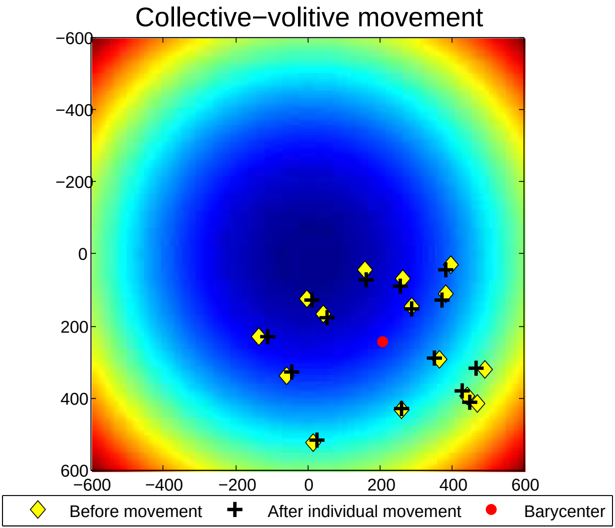
$$\vec{x}(t+1) = \vec{x}(t) \; - \; step_{vol} randu(0,1) \frac{(\vec{x}(t) - \vec{b}(t))}{distance(\vec{x}(t), \vec{b}(t))}$$

When the total weight *decreased* in the current iteration the update is

$$\vec{x}(t+1) = \vec{x}(t) \; + \; step_{vol} randu(0,1) \frac{(\vec{x}(t) - \vec{b}(t))}{distance(\vec{x}(t), \vec{b}(t))}$$

- *distance*() is a function which returns the Euclidean distance between $\vec{x}$ and $\vec{b}$
- $step_{vol}$ is a predetermined step used to control the displacement from/to the barycenter ($step_{vol} = 2 * step_{ind}$ is a good choice)

# Operators (4) – Collective Volitive Movement cont'd



# Outline

1. Basic Fish School Search
2. Update Strategies
   - Weight Update Strategies
   - Non-linear Step Size Decrease Strategy
   - Combined Strategy
3. Evaluation

# New Update Strategies

Contribution of this work [1]: Investigation and comparison of different newly developed update strategies for FSS

---

- Weight update strategies: aim at adjusting weight of each fish in each iteration (***S1, S2***)
- Step size parameter update strategy: non-linear update to the step size parameters $step_{ind}$ and $step_{vol}$ (***S3***)
- Combined strategy: combination of ***S2***, ***S3***, and an additional parameter (***S4***)

[1] Andreas G.K. Janecek and Ying Tan. Feeding the Fish - Weight Update Strategies for the Fish School Search Algorithm. In ICSI'2011: Second International Conference on Swarm Intelligence, pages 553–562. Springer LNCS 6729, 2011

# Outline

Weight Update Strategies

# FSS Pseudo Code – Operators Involving Weight Updates

```
initialize randomly all fish;
while stop criterion is not met do
    for each fish do
        individual movement
        + evaluate fitness function;
        feeding operator;
    end
    for each fish do
        instinctive movement;
    end
    Calculate barycentre;
    for each fish do
        volitive movement;
        evaluate fitness function;
    end
    update step_ind
end
```

# Strategy *S1* – *Linear Decrease of Weights*

## Strategy *S1*

- Weights of all fish are decreased linearly in each iteration
- Decrease by pre-defined factor $\Delta_{lin}$
- After the weight update in Eqn. (1) the weight of all fish is reduced by $w_i = w_i - \Delta_{lin}$
- All weights smaller than 1 are set to 1

Weight update in feeding operator:

$$w_i(t+1) = w_i(t) + \frac{\Delta f(i)}{max(\Delta f)} \tag{1}$$

# Strategy *S2* – *Fitness Based Decrease of Weights*

## Strategy *S2*

- Fish in poor regions will loose weight more quickly
- Let $f(\vec{x})$ be a vector containing the fitness values of all fish
- Weight of fish will be decreased by $\Delta\vec{f}_{fit\,based} = normalize(f(\vec{x}))$
- *Normalize()* is a function that scales $f(\vec{x})$ in the range $[0, 1]$
- Set $w_i = w_i - \Delta\vec{f}_{fit\,based}, \forall w_i < 1 : w_i = 1$

## Scaling in order to improve results

- $\Delta\vec{f}_{fit\,based}$ needs to be scaled by a constant $c_{fit}$ (between 3 and 5)
- $\Delta\vec{f}_{fit\,based} = (\Delta\vec{f}_{fit\,based}.^2)/c_{fit}$

# Strategy *S2* – *Fitness Based Decrease of Weights*

## Strategy *S2*

- Fish in poor regions will loose weight more quickly
- Let $f(\vec{x})$ be a vector containing the fitness values of all fish
- Weight of fish will be decreased by $\Delta\vec{f}_{fit\,based} = normalize(f(\vec{x}))$
- *Normalize()* is a function that scales $f(\vec{x})$ in the range $[0,1]$
- Set $w_i = w_i - \Delta\vec{f}_{fit\,based}, \forall w_i < 1 : w_i = 1$

## Scaling in order to improve results

- $\Delta\vec{f}_{fit\,based}$ needs to be scaled by a constant $c_{fit}$ (between 3 and 5)
- $\Delta\vec{f}_{fit\,based} = (\Delta\vec{f}_{fit\,based}.^2)/c_{fit}$

# Outline

Non-linear Step Size Decrease Strategy

# FSS Pseudo Code – Operators Involving Step Size Parameters

```
initialize randomly all fish;
while stop criterion is not met do
    for each fish do
        individual movement
        + evaluate fitness function;
        feeding operator;
    end
    for each fish do
        instinctive movement;
    end
    Calculate barycentre;
    for each fish do
        volitive movement;
        evaluate fitness function;
    end
    update step_ind
end
```

# Revision: $step_{ind}$ and $step_{vol}$

- Individual movement:

  $n_j(t) = x_j(t) + randu(-1,1) * step_{ind}$

- Collective-volitive movement:

  $\vec{x}(t+1) = \vec{x}(t) - step_{vol}\, randu(0,1) \frac{(\vec{x}(t) - \vec{b}(t))}{distance(\vec{x}(t), \vec{b}(t))}$

- *Linear* decrease of $step_{ind}$:

  $step_{ind}(t+1) = step_{ind}(t) - \frac{step_{ind\ initial} - step_{ind\ final}}{number\ of\ iterations}$

# *S3* – Non-linear Decrease of $step_{ind}$ and $step_{vol}$

## Strategy *S3*

- Non-linear decrease of the step size parameters
- Based on the shape of an ellipse
- Algorithm is forced to converge earlier to the (ideally global) minimum
- Area around the optimum can be searched in more detail

## $step_{vol}$ vs. $step_{ind}$

- Remember that $step_{vol} = 2 * step_{ind}$

# *S3* – Non-linear Decrease of $step_{ind}$ and $step_{vol}$

## Strategy *S3*

- Non-linear decrease of the step size parameters
- Based on the shape of an ellipse
- Algorithm is forced to converge earlier to the (ideally global) minimum
- Area around the optimum can be searched in more detail

## $step_{vol}$ vs. $step_{ind}$

- Remember that $step_{vol} = 2 * step_{ind}$

# S3 – Non-linear Decrease of $step_{ind}$ and $step_{vol}$

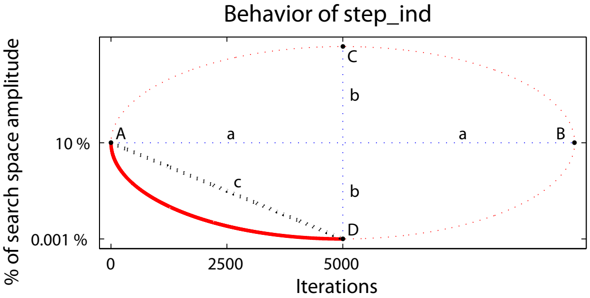



Figure: Linear and non-linear decrease of *step_ind* and *step_vol*

- Bold red curve: new non-linear step size parameter $step_{ind\ nonlin}$
- Dotted line ("c"): linear decrease of $step_{ind}$

# *S3* – Non-linear Decrease of $step_{ind}$ and $step_{vol}$

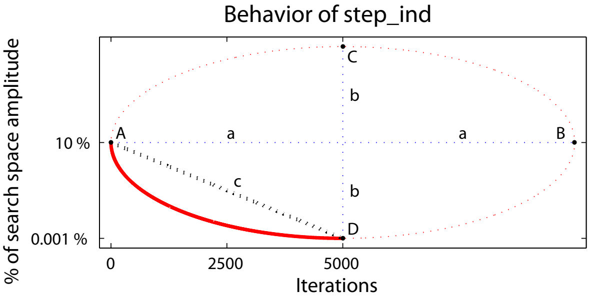
- Let $a$ be the *maximum* number of iterations
- Let $b$ be the distance between $step_{ind\ initial}$ and $step_{ind\ final}$
- Let $t$ be the number of the current iteration

In each iteration $step_{ind\ nonlin}(t)$ is calculated by

$$step_{ind\ nonlin}(t) = step_{ind\ initial} - sqrt\left[(1 - t^2/a^2) * b^2\right]$$

- Derived from the canonical ellipse equation $x^2/a^2 + y^2/b^2 = 1$
- $x$ is replaced with $t$, $y$ is replaced with $step_{step_{ind\ nonlin}(t)}$

# S3 – Non-linear Decrease of $step_{ind}$ and $step_{vol}$




**Basic FSS**: $step_{ind}(t+1) = step_{ind}(t) - \frac{step_{ind\ initial} - step_{ind\ final}}{number\ of\ iterations}$

**Non-linear**: $step_{nonlinear}(t) = step_{ind\ initial} - sqrt\left[(1 - t^2/a^2) * b^2\right]$

# Outline

Combined Strategy

# *S4* – Combination of *S2*, *S3* and a Dilation Multiplier

## Strategy *S4*

- Combines ***S2***, ***S3*** and newly introduced dilation multiplier $c_{dil}$
- Allows to cover a bigger area of the search space when a dilation occurs in the collective volitive movement (i.e. when the total weight of the school decreased in the current iteration)
- The general idea behind the dilation multiplier is to help the algorithm to jump out of local minima
- ***S2*** and ***S3*** are applied in every iteration
- In case of dilation all weights are reset to their initial weight

# *S4* – Combination of *S2*, *S3* and a Dilation Multiplier

```
while stop criterion is not met do
    apply basic FSS operators including S2 and S3;
    %contraction of dilation?
    if (w(t) > w(t − 1)) then
        %contraction:  standard collective-volitive movem.
    else
        %in case of dilation apply the following movement.
        w⃗(t) = 1;
        x⃗(t+1) = x⃗(t) + c_dil * step_vol * randu(0,1) * (x⃗(t) − b⃗(t)) / distance(x⃗(t), b⃗(t))
    end
end
```

$$\vec{w}(t) = 1;$$

$$\vec{x}(t+1) = \vec{x}(t) + c_{dil} * step_{vol} * randu(0,1) * \frac{(\vec{x}(t) - \vec{b}(t))}{distance(\vec{x}(t), \vec{b}(t))}$$

# Outline

1. Basic Fish School Search
2. Update Strategies
3. Evaluation
    - Fitness per Iteration
    - Final Results

# Outline

Fitness per Iteration

# Strategy *S1* – *Linear Decrease of Weights*

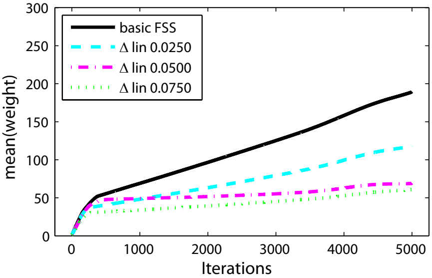
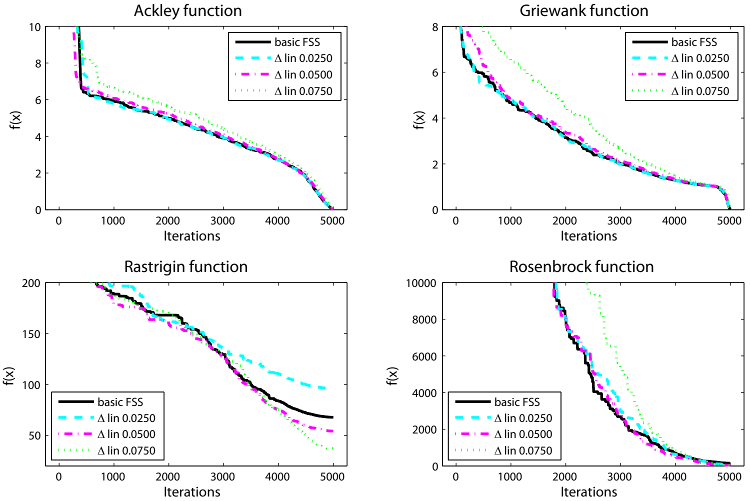



Figure: Average (mean) weight of all fish per iteration

- Weights are decreased linearly: $w_i = w_i - \Delta_{lin}$
- $\Delta_{lin}$ ranges from 0.0125 to 0.0750
- Abbreviated as "Δ lin 0.0XXX" in the figure

# Strategy *S1* – *Linear Decrease of Weights*



# Strategy *S2* – *Fitness based Decrease of Weights*

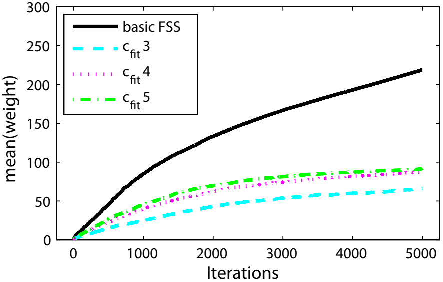
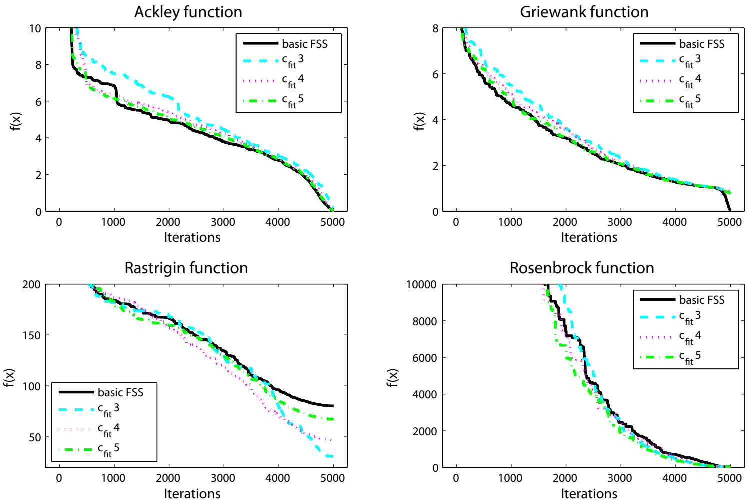



Figure: Average (mean) weight of all fish per iteration

- $\Delta \vec{f}_{fit\,based} = normalize(f(\vec{x}))$
- Scaling: $\Delta \vec{f}_{fit\,based} = (\Delta \vec{f}_{fit\,based}.^2)/c_{fit}$ (abbreviated as "$c_{fit}X$")
- $w_i = w_i - \Delta \vec{f}_{fit\,based}$

# Strategy ***S2*** – *Fitness based Decrease of Weights*



# S3 – Non-linear Decrease of $step_{ind}$ and $step_{vol}$

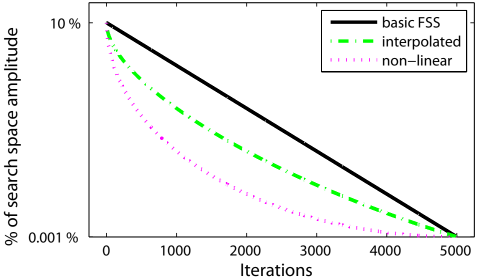
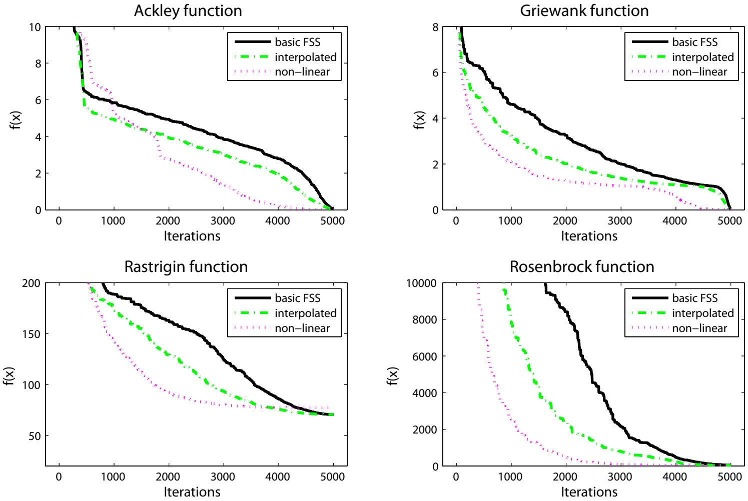



Figure: Behavior of $step_{ind}$

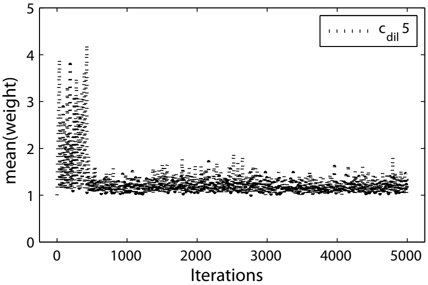
- $step_{nonlinear}(t)$ is abbreviated as “non-linear”
- “Interpolated”: interpolation of “basic FSS” and “non-linear”

  $step_{interpol}(t) = step_{ind}(t) - [step_{ind}(t) - step_{nonlinear}(t)]/2$

# *S3 – Non-linear Decrease of $step_{ind}$ and $step_{vol}$*

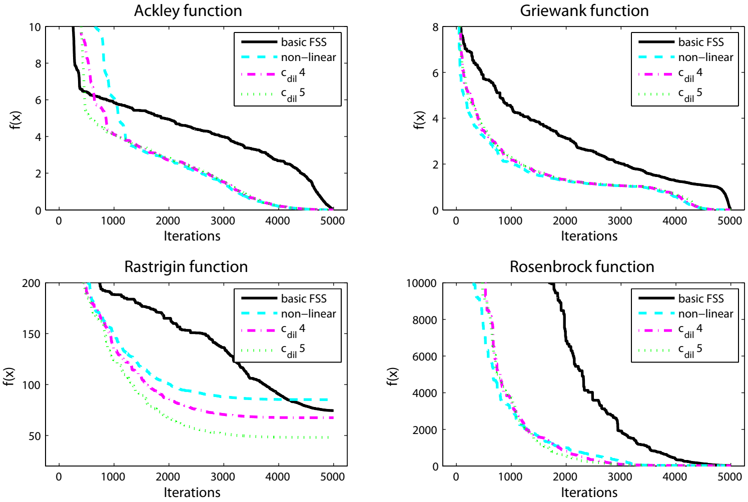


# *S4* – *Combination of* ***S2***, ***S3*** *and a Dilation Multiplier*




- Comparison to basic FSS and "non-linear" from strategy ***S3***
- Dilation multiplier $c_{dilation\,param}$ is abbreviated as "dil.mult.X"
- Since the weight of all fish is reset to 1 if a dilation occurs, the average (mean) weight per iteration is relatively low

# *S4 – Combination of S2, S3 and a Dilation Multiplier*



# Outline

Final Results

# Final Fitness after given Number of Iterations

Table: Mean value and standard deviation (in small font under mean vaule) for 15 trials after 5 000 iterations for the five benchmarks functions

| Function | basic FSS | ***S1*** | ***S2*** | ***S3*** | ***S4*** |
|---|---|---|---|---|---|
| $F_{Ackley}(\vec{x})$ | 0.0100<br>*0.0019* | 0.0100<br>*0.0023* | 0.1270<br>*0.0043* | **0.0007**<br>*6.7e-05* | **0.0007**<br>*5.3e-05* |
| $F_{Griewank}(\vec{x})$ | 0.0233<br>*0.0098* | 0.0172<br>*0.0061* | 0.7501<br>*0.1393* | 0.0058<br>*0.0048* | **3.2e-05**<br>*5.7e-06* |
| $F_{Rastrigin}(\vec{x})$ | 67.126<br>*15.834* | 36.879<br>*8.0181* | **30.745**<br>*11.801* | 70.443<br>*19.465* | 48.156<br>*13.780* |
| $F_{Rosenb.}(\vec{x})$ | 27.574<br>*1.2501* | 28.498<br>*1.3876* | 26.277<br>*2.3244* | **22.775**<br>*2.5801* | 23.718<br>*2.5353* |
| Runtime | 1.0 | × 1.002 | × 1.004 | × 1.014 | × 1.024 |

# Final Fitness after given Number of Iterations

Table: Mean value and standard deviation (in small font under mean vaule) for 15 trials after 5 000 iterations for the five benchmarks functions

| Function | basic FSS | ***S1*** | ***S2*** | ***S3*** | ***S4*** |
|---|---|---|---|---|---|
| $F_{Ackley}(\vec{x})$ | 0.0100 *0.0019* | 0.0100 *0.0023* | 0.1270 *0.0043* | **0.0007** *6.7e-05* | **0.0007** *5.3e-05* |
| $F_{Griewank}(\vec{x})$ | 0.0233 *0.0098* | 0.0172 *0.0061* | 0.7501 *0.1393* | 0.0058 *0.0048* | **3.2e-05** *5.7e-06* |
| $F_{Rastrigin}(\vec{x})$ | 67.126 *15.834* | 36.879 *8.0181* | **30.745** *11.801* | 70.443 *19.465* | 48.156 *13.780* |
| $F_{Rosenb.}(\vec{x})$ | 27.574 *1.2501* | 28.498 *1.3876* | 26.277 *2.3244* | **22.775** *2.5801* | 23.718 *2.5353* |
| Runtime | 1.0 | × 1.002 | × 1.004 | × 1.014 | × 1.024 |

# Conclusion – Update Steps for FSS

Weight update strategies ***S1***, ***S2***

- Linear and fitness-based decrease of weights

⇒ Significant improve of final results of Rastrigin function

Non-linear decrease of step size parameters ***S3***

- Force algorithm to converge earlier to the (ideally global) minimum
- Area around the optimum can be searched in more detail

⇒ Significant improvement in terms of fitness per iteration

Combined Strategy ***S4***

- ***S2***, ***S3*** and a dilation multiplier
- Allows to cover a bigger area of the search space when a dilation occurs in collective-volitive movement

⇒ Overall best results achieved with this strategy

# Conclusion – Update Steps for FSS

Weight update strategies ***S1***, ***S2***

- Linear and fitness-based decrease of weights

⇒ Significant improve of final results of Rastrigin function

Non-linear decrease of step size parameters ***S3***

- Force algorithm to converge earlier to the (ideally global) minimum
- Area around the optimum can be searched in more detail

⇒ Significant improvement in terms of fitness per iteration

Combined Strategy ***S4***

- ***S2***, ***S3*** and a dilation multiplier
- Allows to cover a bigger area of the search space when a dilation occurs in collective-volitive movement

⇒ Overall best results achieved with this strategy

# Conclusion – Update Steps for FSS

Weight update strategies ***S1***, ***S2***

- Linear and fitness-based decrease of weights

⇒ Significant improve of final results of Rastrigin function

Non-linear decrease of step size parameters ***S3***

- Force algorithm to converge earlier to the (ideally global) minimum
- Area around the optimum can be searched in more detail

⇒ Significant improvement in terms of fitness per iteration

Combined Strategy ***S4***

- ***S2***, ***S3*** and a dilation multiplier
- Allows to cover a bigger area of the search space when a dilation occurs in collective-volitive movement

⇒ Overall best results achieved with this strategy

# Further References

- For more information about Fish School Search please visit the official Fish School Search web site by Prof. C. Bastos Filho and Prof. F. Lima Neto at the University of Pernambuco

  http://www.fbln.pro.br/fss

- Other research groups around the world working on FSS can also be found at this web site

  http://www.fbln.pro.br/fss/links.htm

- Implementations/Versions: Implementations of FSS in different programming languages can be downloaded from

  http://www.fbln.pro.br/fss/versions.htm

andreas.janecek@univie.ac.at

**Complete paper:** Andreas G.K. Janecek and Ying Tan. Feeding the Fish - Weight Update Strategies for the Fish School Search Algorithm. In ICSI'2011: Second International Conference on Swarm Intelligence, pages 553–562. Springer LNCS 6729, 2011

Download link: `http://www.cil.pku.edu.cn/publications/papers/2011/ICSI2011Andreas2.pdf`