Self-organizing Systems

Course 9: Cat Swarm Optimization

Sl. dr. ing. Andrei Horia Mogoș

Contents

1. Cat Swarm Optimization

- **1.1 Initial discussion**
- 1.2 Seeking mode
- 1.3 Tracing mode
- 1.4 The CSO algorithm
- 1.5 Experimental results
- 1.6 Conclusions

1.1. Initial discussion

- Cat Swarm Optimization (CSO) is a swarm intelligence method proposed in (Chu, *et al.*, 2006), (Chu, Tsai, 2007).
- CSO algorithm is inspired from the behavior of cats.
- In nature there are about 30 species of felines and all share the same behavior patterns.
- Thus, CSO algorithm can also be seen as a way of simulating felines' behavior.

1.1. Initial discussion

- By observing the behavior of cats, one can find to types of behavior:
  - Most of the time a cat is resting, but it analyzes it's surrounding environment
  - A cat has a very high level of alertness
- In the light of these observations, two behavioral patterns are used for the CSO algorithm:
  - **Seeking mode**: models the periods while the cat is resting and it analyses its surrounding environment
  - **Tracing mode**: models the periods when the cat traces targets

1. Cat Swarm Optimization

1.2. Seeking mode

We define the following four factors:

- **seeking memory pool (SMP)**: the size of the memory of the cat. Only SMP points of the environment will be analyzed, for a given position
- **seeking range of the selected dimension (SRD)**: for each selected dimension, the distance between the current position and the next ones cannot exceed the range, defined by SRD; SRD is a value between 0% and 100%
- **counts of dimensions to change (CDC)**: shows how many dimensions will be varied
- **self position consideration (SPC)**: a boolean value that shows whether the current position of the cat is one of the next candidate positions

1.2. Seeking mode

Seeking mode for cat_k is described as follows:

- **Step 1**: Let be $j = SMP$. Make j copies of the present position of cat_k. If $SPC = \text{true}$ then make $j = (SMP - 1)$ and select the current position as one of the candidates.
- **Step 2**: For each copy (there are j copies), randomly select CDC dimensions to be modified; then for each selected dimension d, randomly choose a value v_{new} from $[v_{crt} - SRD \cdot v_{crt}, v_{crt} + SRD \cdot v_{crt}]$, where v_{crt} is the current value for dimension d.
- **Step 3**: Let be $Cand$ the set of candidate points. Compute the fitness values $FS_i, i \in Cand$.

1.2. Seeking mode

Seeking mode for cat_k is described as follows (cont.):

- **Step 4**: If all $FS_i, i \in Cand$ are exactly equal then each candidate point will have the selecting probability equal to 1, otherwise, the selecting probabilities for the candidate points will be computed using equation (1):

$$P_i = \frac{|FS_i - FS_b|}{FS_{max} - FS_{min}}, \ i \in Cand \tag{1}$$

  where FS_{max} is the maximum value and FS_{min} is the minimum value of the set $\{FS_i | i \in Cand\}$. If the goal is to find the maximum solution then $FS_b = FS_{min}$, otherwise, $FS_b = FS_{max}$.

- **Step 5**: According to the selecting probabilities computed in Step 4, randomly pick a point from $Cand$ and move the cat_k to that point.

1. Cat Swarm Optimization

- 1.1 Initial discussion
- 1.2 Seeking mode
- **1.3 Tracing mode**
- 1.4 The CSO algorithm
- 1.5 Experimental results
- 1.6 Conclusions

1.3. Tracing mode

Tracing mode for cat_k is described as follows:

- **Step 1**: For each dimension d, update the velocity $v_{k,d}$ of cat_k as specified in equation (2):

$$v_{k,d} = v_{k,d} + r_1 \cdot c_1 \cdot (x_{best,d} - x_{k,d}), \ d = 1, 2, ..., M \tag{2}$$

  where

  - M is the number of dimensions
  - $x_{best,d}$ is the position (for dimension d) of the best cat
  - $x_{k,d}$ is the position (for dimension d) of the cat_k
  - c_1 is a constant
  - r_1 is a random value in [0,1]

1.3. Tracing mode

Tracing mode for cat_k is described as follows:

- **Step 2**: Let be v_{max} the maximum velocity. For each dimension, adjust $v_{k,d}$ according to the rules:
  - if $v_{k,d} < -v_{max}$ then $v_{k,d} = -v_{max}$
  - if $v_{k,d} > v_{max}$ then $v_{k,d} = v_{max}$
  - Otherwise, $v_{k,d}$ remains unchanged
- **Step 3**: For each dimension d, update the position $x_{k,d}$ of cat_k as specified in equation (3):

$$x_{k,d} = x_{k,d} + v_{k,d} \tag{3}$$

1. Cat Swarm Optimization

- 1.1 Initial discussion
- 1.2 Seeking mode
- 1.3 Tracing mode
- **1.4 The CSO algorithm**
- 1.5 Experimental results
- 1.6 Conclusions

1.4. The CSO algorithm

- The CSO algorithm uses the two behavior modes explained before: Seeking mode and Tracing mode.
- A **mixture ratio (MR)** is used to combine the two behavior modes (MR is between 0% and 100%)
- At a given moment, MR cats of the population will be in Tracing mode and $100\% - MR$ cats will be in the Seeking mode.
- A cat spends most of the time resting and analysing the environment.
- Consequently, the majority of the cats of the population must be in Seeking mode.
- Thus, MR must be a small value (here 2% is used).

1.4. The CSO algorithm

The CSO algorithm is described as follows:

- **Step 1**: Create a population of N cats.
- **Step 2**: Perform the following actions:
  - Randomly put the cats into the solution space (the solution space has M dimensions)
  - Randomly give values to the velocities of the cats in $[-v_{max}, v_{max}]$
  - Randomly choose $MR \cdot N$ cats and put them in Tracing mode; put the other cats in the Seeking mode.
- **Step 3**: For each cat, evaluate (using the fitness function) the fitness value of the current position of the cat; then, memorize the best cat.

1.4. The CSO algorithm

The CSO algorithm is described as follows (cont.):

- **Step 4**: Move the cats: for each cat, apply its current behavior mode (Seeking mode or Tracing mode)
- **Step 5**: Randomly choose $MR \cdot N$ cats and put them in Tracing mode; put the other cats in the Seeking mode.
- **Step 6**: If the stopping criteria are satisfied then stop the program, otherwise repeat Step 3 - Step 5.

1.4. The CSO algorithm

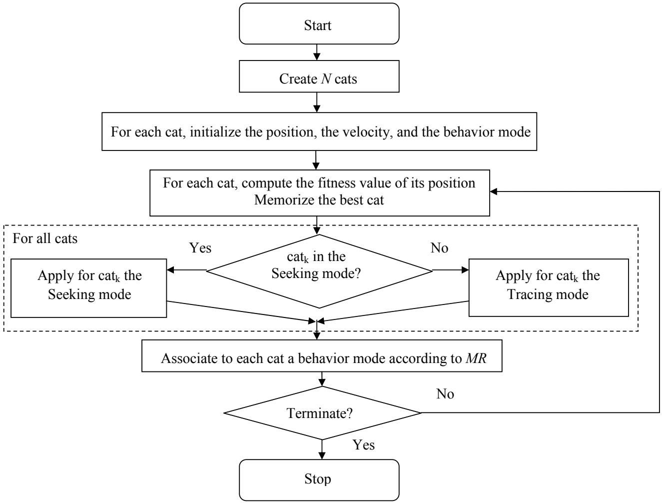

Figure: CSO Process Diagram

1. Cat Swarm Optimization

1.5. Experimental results

The tests are made on six functions:

- $f_1(x) = \sum_{d=1}^{M} [100(x_d - x_{d-1})^2 + (x_{d-1} - 1)^2]$
  (a variant of Rosenbrock function)
- $f_2(x) = \sum_{d=1}^{M} [x_d^2 - 10 \cdot \cos(2\pi x_d)^2 + 10]$
  (a variant of Rastrigin function)
- $f_3(x) = \frac{1}{400} \sum_{d=1}^{M} x_d^2 - \prod_{d=1}^{M} \cos(\frac{x_d}{\sqrt{d}}) + 1$
  (a variant of Griewank function)
- $f_4(x) = 20 + \exp^1 - 20 \cdot \exp^{-0.2\sqrt{\frac{\sum_{d=1}^{M} x_d^2}{M}}} - \exp^{\sum_{d=1}^{M} \frac{\cos(2\pi x_d)}{M}}$
- $f_5(x) = \sum_{d=1}^{M} (x_d^2 - 10\cos(2\pi x_d) + 10)$
- $f_6(x) = \sum_{d=1}^{M} \lfloor x_d + 0.5 \rfloor^2$

1.5. Experimental results

The table presented below, contains the initial ranges for the six test functions.

Test Function	Range
Test Function 1	$x_d \in [15, 30]$
Test Function 2	$x_d \in [2.56, 5.12]$
Test Function 3	$x_d \in [300, 600]$
Test Function 4	$x_d \in [-30, 30]$
Test Function 5	$x_d \in [-5.12, 5.12]$
Test Function 6	$x_d \in [-100, 100]$

Table: Initial ranges for the test functions

These limits are necessary only for the initialization of the algorithms. At later steps of the algorithms, for each dimension, any value is accepted.

1.5. Experimental results

The values / ranges of the parameters for CSO are presented in the next table:

Parameter	Value / Range
SMP	5
SRD	20%
CDC	80%
MR	2%
c_1	2
r_1	[0, 1]

Table: Parameters for CSO

1.5. Experimental results

CSO is compared with PSO and PSO with WF (weighting factor, i.e. inertia weight). The next table contains the parameters values / ranges used for PSO and PSO with WF:

Parameter	Value / Range
Initial Weight	0.9
Final Weight	0.4
c_1	2
c_2	2
r_1	[0, 1]
r_2	[0, 1]

Table: Parameters for PSO and PSO with WF

1.5. Experimental results

The next table contains, for each test function, the maximum velocity used for PSO and PSO with WF:

Test Function	v_{max}
Test Function 1	100
Test Function 2	10
Test Function 3	600
Test Function 4	100
Test Function 5	600
Test Function 6	10

Table: Maximum velocity for PSO and PSO with WF

1.5. Experimental results

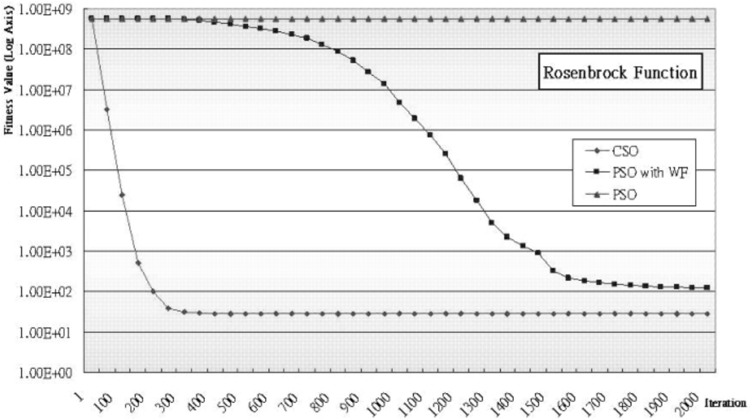

Figure: Results for Rosenbrock function (taken from (Chu, Tsai, 2007))

1.5. Experimental results

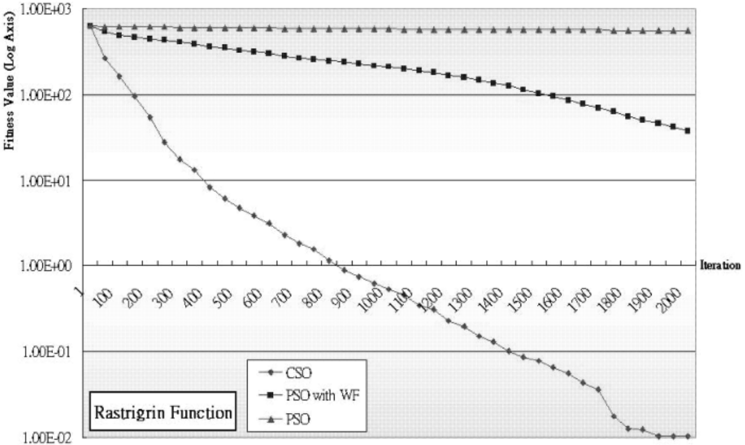

Figure: Results for Rastrigin function (taken from (Chu, Tsai, 2007))

1.5. Experimental results

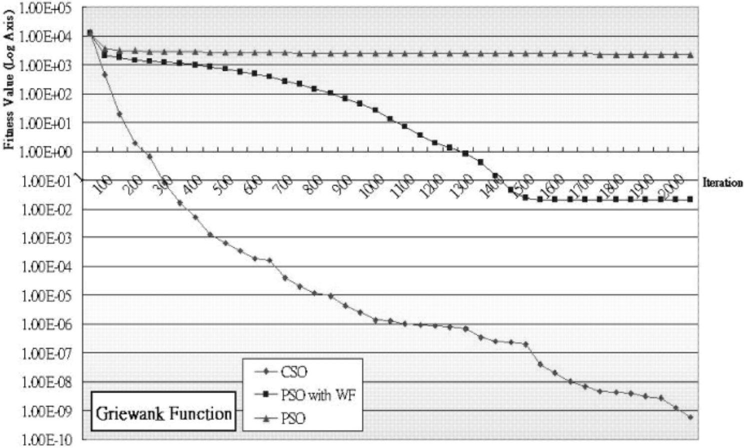

Figure: Results for Griewank function (taken from (Chu, Tsai, 2007))

1.5. Experimental results

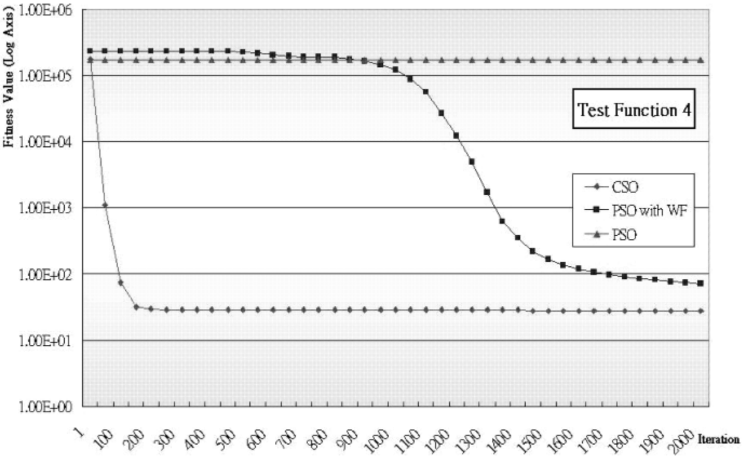

Figure: Results for Test Function 4 (taken from (Chu, Tsai, 2007))

1.5. Experimental results

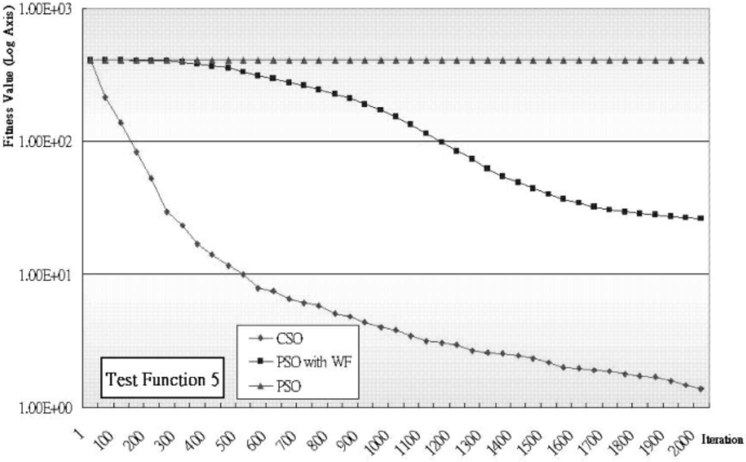

Figure: Results for Test Function 5 (taken from (Chu, Tsai, 2007))

1.5. Experimental results

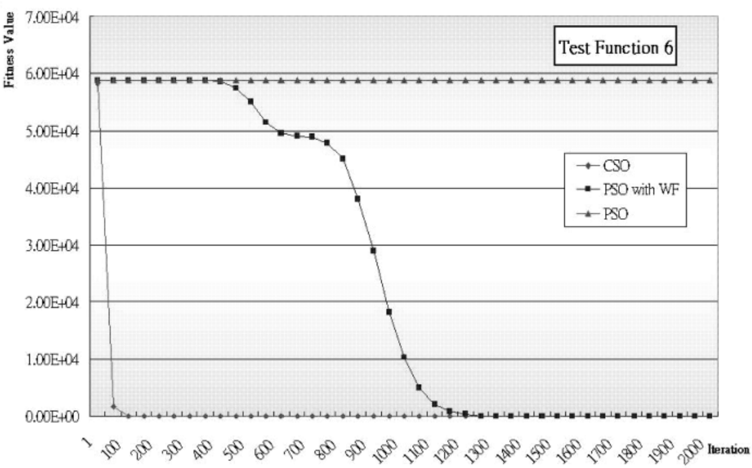

Figure: Results for Test Function 6 (taken from (Chu, Tsai, 2007))

1. Cat Swarm Optimization

1.6. Conclusions

- Cat Swarm Optimization is a new swarm intelligence algorihtm, based on the behavior of cats.
- For the same iteration, CSO consumes more time than PSO and PSO with WF.
- CSO is better (for finding the global optimum) than PSO and PSO with WF, for the six test functions and for the same number of iterations.

2. An Inertia Weighted Cat Swarm Optimization for Clustering

- **2.1 Initial discussion**
- 2.2 An inertia weighted Cat Swarm Optimization
- 2.3 Experimental results
- 2.4 Conclusions

2.1. Initial discussion

- In (Santosa, Ningrum, 2009) the authors use CSO for clustering.
- They use two variants of CSO: standard CSO and an inertia weighted CSO.
- The clustering is performed on four datasets: Iris, Soybean-small, Glass, Balance Scale
- The authors compare the results obtained by the two variants of CSO.
- They also compare the results obtained by inertia weighted CSO with the ones obtained by using K-means and PSO

2. An Inertia Weighted Cat Swarm Optimization for Clustering

- 2.1 Initial discussion
- **2.2 An inertia weighted Cat Swarm Optimization**
- 2.3 Experimental results
- 2.4 Conclusions

2.2 An inertia weighted Cat Swarm Optimization

Remember **Step 1**, of **the Tracing mode**:

- For each dimension d, update the velocity $v_{k,d}$ of cat_k as specified below:

$$v_{k,d} = v_{k,d} + r_1 \cdot c_1 \cdot (x_{best,d} - x_{k,d}), \ d = 1, 2, ..., M$$

where

  - M is the number of dimensions
  - $x_{best,d}$ is the position (for dimension d) of the best cat
  - $x_{k,d}$ is the position (for dimension d) of the cat_k
  - c_1 is a constant
  - r_1 is a random value in [0,1]

2.2 An inertia weighted Cat Swarm Optimization

- **An inertia weighted CSO** is obtained by adding a weight to the inertia term of the velocity formula, i.e. to the current velocity.
- The new formula for the velocity is the following:

$$v_{k,d} = W \cdot v_{k,d} + r_1 \cdot c_1 \cdot (x_{best,d} - x_{k,d}),\ d = 1, 2, ..., M$$

where W is a random value in [0, 1].

2. An Inertia Weighted Cat Swarm Optimization for Clustering

- 2.1 Initial discussion
- 2.2 An inertia weighted Cat Swarm Optimization
- **2.3 Experimental results**
- 2.4 Conclusions

2.3. Experimental results

Dataset	Criteria	CPU time (second)		Clustering Error	
		Mean	*std. deviation*	*Mean (%)*	*std. deviation*
Iris	Original	0.45	0.07	23.34	.119
	Modified	0.64	0.12	17.55	.096
Soybean	Original	0.32	0.04	33.19	.126
	Modified	0.40	0.11	24.60	.094
Glass	Original	1.17	0.22	56.34	.060
	Modified	1.24	0.18	51.63	.049
Balance Scale	Original	2.53	0.72	48.54	.071
	Modified	2.59	0.54	46.80	.060

Table: Clustering results for standard CSO and proposed inertia weighted CSO (taken from (Santosa, Ningrum, 2009))

Remark: For information related to the clustering algorithm, consult (Santosa, Ningrum, 2009).

2.3. Experimental results

CSO Clustering		K-means		PSO Clustering	
Parameter	*Value*	*Parameter*	*Value*	*Parameter*	*Value*
maxiter	50	maxiter	50	maxiter	50
copy	5			w	1
SRD	0.2			c1	2
const1	2			c2	2
r1	[0,1]			r1	[0,1]
velmax	0.9			r2	[0,1]
				velmax	0.9

Table: Parameters for clustering (taken from (Santosa, Ningrum, 2009))

2.3. Experimental results

Data		Method	Time	Error
1	Iris	CSO Clustering	0.64	17.55%
		K-means	0.00	18.30%
		PSO Clustering	0.13	27.43%
2	Soybean	CSO Clustering	0.60	24.19%
		K-means	0.00	27.62%
		PSO Clustering	0.22	35.17%
3	Glass	CSO Clustering	1.24	51.63%
		K-means	0.03	48.99%
		PSO Clustering	0.20	56.36%
4	Balance Scale	CSO Clustering	2.59	46.80%
		K-means	0.13	48.11%
		PSO Clustering	0.49	48.74%

Table: Clustering results for CSO, K-means, and PSO (taken from (Santosa, Ningrum, 2009))

2. An Inertia Weighted Cat Swarm Optimization for Clustering

- 2.1 Initial discussion
- 2.2 An inertia weighted Cat Swarm Optimization
- 2.3 Experimental results
- **2.4 Conclusions**

2.4. Conclusions

- In (Santosa, Ningrum, 2009) an inertia weighted CSO is proposed.
- Inertia weighted CSO works better than standard CSO, on the four datasets.
- Inertia weighted CSO works better than K-means and PSO on three datasets (Iris, Soybean-small, and Balance Scale).
- The running-time for inertia weighted CSO is higher than the ones for K-means and PSO, on the four datasets.

3. Average-Inertia Weighted Cat Swarm Optimization

- **3.1 Initial discussion**
- 3.2 Inertia weighted CSO (ICSO)
- 3.3 Average-inertia weighted CSO (AICSO)
- 3.4 Experimental results
- 3.5 Conclusions

3.1. Initial discussion

- In (Orouskhani, *et al.*, 2011) the authors propose two new variants of the CSO algorithm.
- In the first variant the authors modify the velocity formula, by adding an inertia weight.
- In the second variant the authors modify both the velocity formula and the position formula.
- They compare the two variants with standard CSO on three test functions: Rastrigin function, Griewank function and Ackley function.

3. Average-Inertia Weighted Cat Swarm Optimization

- 3.1 Initial discussion
- **3.2 Inertia weighted CSO (ICSO)**
- 3.3 Average-inertia weighted CSO (AICSO)
- 3.4 Experimental results
- 3.5 Conclusions

3.2. Inertia weighted CSO (ICSO)

Remember **Step 1**, of **the Tracing mode**:

- For each dimension d, update the velocity $v_{k,d}$ of cat_k as specified below:

$$v_{k,d} = v_{k,d} + r_1 \cdot c_1 \cdot (x_{best,d} - x_{k,d}),\ d = 1, 2, ..., M$$

where

  - M is the number of dimensions
  - $x_{best,d}$ is the position (for dimension d) of the best cat
  - $x_{k,d}$ is the position (for dimension d) of the cat_k
  - c_1 is a constant
  - r_1 is a random value in [0,1]

3.2. Inertia weighted CSO (ICSO)

- **An inertia weighted CSO** is obtained by adding a weight to the inertia term of the velocity formula, i.e. to the current velocity.
- The new formula for the velocity is the following:

$$v_{k,d} = w \cdot v_{k,d} + r_1 \cdot c_1 \cdot (x_{best,d} - x_{k,d}),\ d = 1, 2, ..., M$$

where w is at the begining 0.9 and it is decreased until 0.4.

- The formula for the position remains the same (compared with standard CSO):

$$x_{k,d} = x_{k,d} + v_{k,d}$$

3. Average-Inertia Weighted Cat Swarm Optimization

3.3. Average-inertia weighted CSO (AICSO)

- Starts from Inertia weighted CSO (ICSO)
- Modifies the formula of the position

3.3. Average-inertia weighted CSO (AICSO)

- The formula for the velocity is the following (the same formula used for ICSO):

$$v_{k,d} = w \cdot v_{k,d} + r_1 \cdot c_1 \cdot (x_{best,d} - x_{k,d}), \; d = 1, 2, ..., M$$

where w is at the begining 0.9 and it is decreased until 0.4.

- The formula for the position is modified as presented below:

$$x_{k,d} = x_{k,d} + v_{k,d}$$

$$x_{i+1} = \frac{(x_{i+1} + x_i)}{2} + \frac{(v_{i+1} + v_i)}{2}$$

where x_{i+1} and v_{i+1} are the position and the velocity computed at the current step, while x_i and v_i are the position and the velocity computed at the previous step.

3. Average-Inertia Weighted Cat Swarm Optimization

3.4. Experimental results

The test functions are presented below:

- $f_1(x) = \sum_{d=1}^{M} [x_d^2 - 10 \cdot \cos(2\pi x_d)^2 + 10]$
  (a variant of Rastrigin function)
- $f_2(x) = \frac{1}{4000} \sum_{d=1}^{M} x_d^2 - \prod_{d=1}^{M} \cos(\frac{x_d}{\sqrt{d}}) + 1$
  (a variant of Griewank function)
- $f_3(x) = 20 + \exp^1 - 20 \cdot \exp^{-0.2\sqrt{\frac{\sum_{d=1}^{M} x_d^2}{M}}} - \exp^{\sum_{d=1}^{M} \frac{\cos(2\pi x_d)}{M}}$
  (the Ackley function)

3.4. Experimental results

Parameter	Value / Range
SMP	5
SRD	20%
CDC	80%
MR	2%
c_1	2.05
r_1	[0, 1]
w	[0.4, 0.9]

Table: Parameters for ICSO and AICSO

3.4. Experimental results

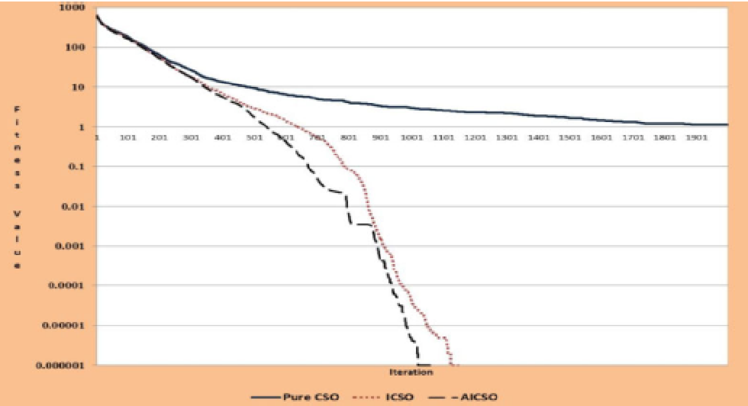

Figure: Results for Rastrigin Function (taken from (Orouskhani, *et al.*, 2011))

3.4. Experimental results

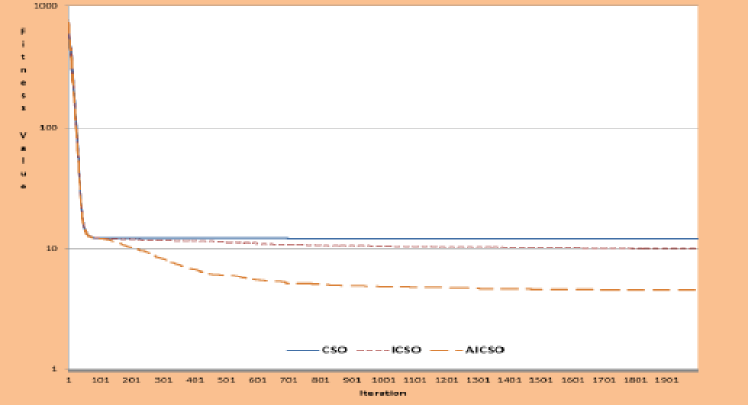

Figure: Results for Griewank Function (taken from (Orouskhani, *et al.*, 2011))

3.4. Experimental results

Figure: Results for Ackley Function (taken from (Orouskhani, *et al.*, 2011))

3. Average-Inertia Weighted Cat Swarm Optimization

- 3.1 Initial discussion
- 3.2 Inertia weighted CSO (ICSO)
- 3.3 Average-inertia weighted CSO (AICSO)
- 3.4 Experimental results
- **3.5 Conclusions**

3.5. Conclusions

- The authors propose two new versions (ICSO and AICSO) of the Cat Swarm Optimization algorithm.
- The experiments, made on three test functions, show that for the analyzed functions, AICSO has better performances than standard CSO.
- For the first two test functions, the experimental results show that AICSO is better than ICSO.

References

1. S. C. Chu, P. W. Tsai, J. S. Pan, "Cat Swarm Optimization", Q. Yang and G. Webb (Eds.): PRICAI 2006, LNAI 4099, 2006, pp. 854 - 858
2. S. C. Chu, P. W. Tsai, "Computational Intelligence Based on the Behavior of Cats", International Journal of Innovative Computing, Information and Control, vol. 3, no. 1, 2007, pp. 163 - 173
3. M. Orouskhani, M. Mansouri, M. Teshnehlab, "Average - Inertia Weighted Cat Swarm Optimization", Y. Tan et al. (Eds.): ICSI 2011, Part I, LNCS 6728, 2011, pp. 321 - 328
4. B. Santosa, M. K. Ningrum, "Cat Swarm Optimization for Clustering", Proceedings of 2009 International Conference of Soft Computing and Pattern Recognition, December 4 - 7, 2009, Malacca, Malaysia, pp. 54 - 59