Bird Flock in Turbulent Atmosphere

José Carlos Becceneri
Laboratory for Computing and Applied Mathematics (LAC)
National Institute for Space Research (INPE), São José dos Campos, SP, Brazil.
becce@lac.inpe.br

Stephan Stephan
Laboratory for Computing and Applied Mathematics (LAC)
National Institute for Space Research (INPE), São José dos Campos, SP, Brazil.
stephan@lac.inpe.br

Haroldo Fraga de Campos Velho
Laboratory for Computing and Applied Mathematics (LAC)
National Institute for Space Research (INPE), São José dos Campos, SP, Brazil.
haroldo@lac.inpe.br

Eduardo Fávero Pacheco da Luz
Graduation Program in Applied Computing (CAP)
National Institute for Space Research (INPE), São José dos Campos, SP, Brazil.
eduardo.luz@lac.inpe.br

Antonio José Silva Neto
Instituto de Pesquisas do Rio de Janeiro/UERJ
ajsneto@iprj.uerj.br

Abstract

This work proposes an enhancement to the optimization technique that mimics the flight of a flock of birds, the Particle Swarm Optimization. In the original form of the algorithm, each bird represents a candidate solution and updates its position in the search space from the previous best evaluated positions obtained by itself and by the flock. The proposed innovation is given by the addition of an atmospheric turbulence that affects in an independent, random and sporadic way the flight of each bird of the flock. Optimization tests were performed, using standard real functions, in order to check the new algorithm. It can be concluded that the addition of turbulence is effective, in comparison to the original algorithm, to escape from local minima and to reach better solutions. In addition, the new algorithm is more robust concerning the choice of tuning parameters that balance the influence of the past better positions.

Keywords: Particle Swarm Optimization, flock-of-birds algorithm, function minimization, turbulence.

1. Introduction

This work proposes an enhancement to the optimization technique that mimics the flight of a flock of birds, the Particle Swarm Optimization (PSO). The original version of the algorithm was proposed in 1995 [4]. In the PSO, a flock of birds is associated to the particle swarm and the algorithm tries to mimic the behavior of the birds that compose the flock. Each bird is associated to a candidate solution and updates its position in the search space from the previous best evaluated positions obtained by itself and by the entire flock.

The proposed enhancement to the PSO algorithm is the addition of a new stochastic component to simulate the atmospheric wind, which is always subjected to turbulence [7]. This turbulent component affects in an independent, random and sporadic way the flight of each bird of the flock. We denoted the new algorithm by PSO-T (Particle Swarm Optimization with Turbulence).

Optimization tests were performed, using standard real functions, in order to compare the new PSO-T implementation to a standard PSO one. It can be concluded that the addition of turbulence is effective, in comparison to the original algorithm, to escape from local minima and to reach better solutions. In addition, the new algorithm is more robust concerning the choice of tuning parameters that balance the influence of the past better positions.

The next section is about PSO concepts and the introduction of turbulence. Section 3 discusses the minimization of real functions and describes the standard functions that were employed in the tests, while section 4 discusses the results. Finally, section 5 includes some discussions of the obtained results and proposes future works.

2. The flock of birds paradigm and atmospheric turbulence

One of the main streams in artificial life is to understand how real world animals behave as part of a swarm and to try to mimic this behavior in an algorithm. Some aspects of such behavior must be abstracted in order to obtain rules that are feasible to be implemented in an algorithm. Even when the individual behavior is simple, the collective behavior can be very complex. This is the case of the PSO.

Boyd e Richerson [2] have studied the decision making process in human beings and observed that decisions are taken based on the personal experience, but also on the neighbors' experience. This feature was exploited in the PSO algorithm and applied to the behavior of the birds. It is assumed that the behavior of the flock is a consequence of the effort of each bird in keeping an optimal distance from the neighboring birds.

The aesthetical choreography of a flock of birds was studied by zoology and computer science researchers in order to know what are the rules that provide for the synchronous flight of the flock even subjected to successive changes of direction.

In the PSO, a flock of birds is represented in a n-dimensional search space. The position of each agent/bird *i* in iteration *k* is given by its vector of Cartesian coordinates $\mathbf{X}_i^k$. At every iteration, that corresponds to an unitary amount of time, the flock of birds evolve as a consequence of the update of the positions of each bird. The update of position of agent/bird *i* is calculated using its current velocity vector $\mathbf{V}_i^k$, which is also updated at every iteration as a function of its previous position $\mathbf{X}_i^{k-1}$ and velocity $\mathbf{V}_i^{k-1}$.

The position of each bird represents a possible solution in the search space. The evaluation of each bird is performed at every iteration by means of an objective function F($\mathbf{X}$). Each bird stores its best position $\mathbf{X}_i^{pbest}$, that corresponds to the better evaluation obtained by itself. This information is due to its own experience. Every bird also knows the best evaluation obtained by the flock until the moment, $\mathbf{X}^{gbest}$, that correponds to the experience of the group. At every iteration, the velocity vector $\mathbf{V}_i^{k-1}$ of each bird *i* is updated in function of the following variables:

- its previous position $\mathbf{X}_i^{k-1}$
- its previous velocity $\mathbf{V}_i^{k-1}$
- the distance vector defined by its previous position and its $\mathbf{X}_i^{pbest}$
- the distance vector defined by its previous position and flock's $\mathbf{X}^{gbest}$

The new (current) position $\mathbf{X}_i^k$ is defined by applying the current velocity operator to previous position $\mathbf{X}_i^{k-1}$. Actually, for a unitary time step, this is equivalent to add this velocity to the previous position in order to obtain the current position (see eq. (4)).

In the PSO, the following equation defines the current velocity of each bird:

$$V_i^k = c_1.V_i^{k-1} + c_2.rand_1\,(\mathbf{X_i^{pbest}} - \mathbf{X_i^{k-1}}) + c_3.rand_2\,(\mathbf{X^{gbest}} - \mathbf{X_i^{k-1}}) \quad (1)$$

where $rand_1$ and $rand_2$ are random numbers between 0 and 1 and the positive real numbers, denoted learning parameters, must be chosen:

c_1: parameter that express the trust of the bird in itself;
c_2: parameter that express the trust of the bird in its experience;
c_3: parameter that express the trust of the bird in the experience of the flock.

The above learning parameters c_2 and c_3, weight the stochastic accelerations towards positions $\mathbf{X_i^{pbest}}$ and $\mathbf{X^{gbest}}$, respectively [5]. In terms of behavior, the parameter c_2 represents the cognitive factor associated to its best former experience, while the parameter c_3 represents the social factor associated to the best former experience of the group. It is common to assign the same value to these two parameters [3] e [6].

The contribution of this work is the addition of an atmospheric turbulence, which affects in an independent, random and sporadic way the flight of each bird of the flock in the PSO-T (Particle Swarm Optimization with Turbulence). In order to simulate this turbulence, it is required to define how the turbulence will affect the birds of the flock. A new real parameter (q_0), with value between 0 and 1, is used by a roulette scheme: at every iteration a random number is generated and compared to q_0 in order to include or not turbulence in the flight of all birds of the flock. Additionally, a turbulence intensity parameter *iturb* is defined in function of the best evaluation of the flock $F(\mathbf{X^{gbest}})$ and of the number of the elapsed number of iterations with no improvement of this evaluation:

$$iturb = [(c_1+c_2+c_3)/\,F(\mathbf{X^{gbest}})] * [1.0/(nit-nit\_improv)] \quad (2)$$

where,

c_1, c_2, c_3 are the learning parameters of equation (1),
nit: is the iteration number,
nit_improv: is the previous iteration in which $F(\mathbf{X^{gbest}})$ improved.

In the presence of turbulence, in a given iteration, each velocity component of bird i is multiplied by the intensity parameter *iturb* and also by a random number between 0 and 1 ($rand_i$). A new random number is generated for each component of the velocity of each bird. In this way, the turbulence acts as a stochastic forcing, causing a perturbation in each of the n components of velocity $\mathbf{V}_i^k$ of bird i, as follows:

$$v_i^k = v_i^k * iturb * rand_i; \quad (3)$$

where $\mathbf{V}_i^k = [(v_i^k)^1\ (v_i^k)^2 \ldots\ (v_i^k)^n]$. As mentioned before, the position of each bird is updated by the following equation, for a unitary step of time:

$$\mathbf{X}_i^k = \mathbf{X}_i^{k-1} + \mathbf{V}_i^k \quad (4)$$

A general description of the proposed **PSO-T** algorithm follows.

Step 1: Setting of initial conditions for the flock; for each bird, the position ($\mathbf{X}_i^0$) and velocity ($\mathbf{V}_i^0$), are randomly generated, given suitable ranges;

Step 2: Evaluation of the objective function $F(\mathbf{X})$ for each bird of the flock; the positions $\mathbf{X_i^{pbest}}$ and $\mathbf{X^{gbest}}$ are eventually updated;

Step 3: Update of the velocities of each bird of the flock; the velocity of a bird is updated using eq. (1), obtaining $\mathbf{V}_i^k$ and turbulence is applied according to the roulette scheme that uses the parameter q_0; if there is turbulence, eqs. (2) and (3) are used to independently perturb the velocities of each bird of the flock in that iteration;

Step 4: Update of the positions of each bird of the flock using eq. (4), in order to obtain the new positions $\mathbf{X}_i^k$;

Step 5: Check of the stopping criteria; if it is not verified, return to step 2 for the next iteration.

The stopping criteria can be defined in a suitable manner. In this work, it is employed a limit processing time in order to end the algorithm. Other options include a limit number of iterations or a threshold to be reached by the objective function – this would require a careful choice of the threshold to avoid a never-ending algorithm.

3. Function Minimization Tests

In order to check the influence of the addition of turbulence to the original PSO algorithm, we choose to test the new PSO-T in the minimization of real functions. Four well-known functions were minimized: Rosenbrock, Schwefel, Rastrigin e Griewank. These functions are defined at http://www.geatbx.com/docu/fcnindex-01.html as of late May, 2009.

The number of variables of each function can be set and in the present minimization tests we employed 5 variables and a processing time limit as stopping criteria. The search space was limited in a suitable manner for each function, employing intervals that are commonly found in the optimization literature. Table 1 shows, for each function the optimal value and corresponding values of the variables ($\mathbf{X}_i$), as well as the considered domain (“interval” column).

Function	**F_{min} optimal**	**optimal X_{min} (i=1, ...,5)**	**interval (i=1, ...,5)**
Griewank	0	$x_i = 0$	$x_i \in [-600, 600]$
Rastrigin	0	$x_i = 0$	$x_i \in [-5.12, 5.12]$
Rosenbrock	0	$x_i = 1$	$x_i \in [-5.12, 5.12]$
Schwefel	-2094.3225880493	$x_i = 420$	$x_i \in [-500, 500]$

Table 1: Optimal values F_{min}, corresponding $\mathbf{X}_{min}$ and limits for the considered functions.

Additionally, the PSO-T algorithm requires the setting of the following parameters:

- **np**: number of birds of the flock;
- **T_limit:** limit processing time;
- **Seed**: chosen seed for the random number generator;
- **q_0:** parameter associated to the roulette scheme that defines presence/absence of turbulence in a given iteration.

Next, it is presented some test cases. In each test, 1600 birds were utilized, during 1800 seconds. For each function, 25 test cases were executed, each one with a different seed to generate random numbers. The tables show results for different values of c_1, c_2 and c_3. The parameter $q_0 = 0.2$ in all cases.

		PSO			PSO-t		
		Mean	StdDev	Min	Mean	StdDev	Min
Griewank	Cost	0,524	0,126	0,323	0,010	0,007	0,000
	Best Iteraction	173,960	240,706	6,000	29801,360	36951,634	238,000
Rastrigin	Cost	0,255	0,308	0,063	0,000	0,000	0,000
	Best Iteraction	357,840	170,850	127,000	786,680	900,980	196,000
Rosenbrock	Cost	1,908	0,823	0,181	0,398	0,392	0,000
	Best Iteraction	213,280	200,530	26,000	15067,640	16799,088	1768,000
Schwefeld	Cost	-2055,045	65,750	-2094,725	-1949,630	89,717	-2094,914
	Best Iteraction	738,760	474,107	129,000	1439,960	4307,529	161,000

Table 2: $c_1 = 1.0$, $c_2 = 0.1$, $c_3 = 0.1$

		PSO			PSO-t		
		Mean	StdDev	Min	Mean	StdDev	Min
Griewank	Cost	0,567	0,136	0,396	0,012	0,010	0,000
	Best Iteraction	72,750	44,814	31,000	27765,680	62932,680	194,000
Rastrigin	Cost	0,533	0,703	0,017	0,000	0,000	0,000
	Best Iteraction	218,280	143,549	34,000	298,000	241,807	132,000
Rosenbrock	Cost	2,289	0,756	0,954	0,022	0,112	0,000
	Best Iteraction	120,720	137,849	9,000	4964,760	14466,394	1121,000
Schwefeld	Cost	-1978,812	100,806	-2094,685	-1962,264	98,620	-2094,914
	Best Iteraction	222,480	177,716	52,000	852,400	2381,515	134,000

Table 3: $c_1 = 1.0$, $c_2 = 0.2$, $c_3 = 0.2$

		PSO			PSO-t		
		Mean	StdDev	Min	Mean	StdDev	Min
Griewank	Cost	0,000	0,000	0,000	0,000	0,000	0,000
	Best Iteraction	1673,120	3442,228	211,000	1661,160	1806,964	224,000
Rastrigin	Cost	0,000	0,000	0,000	0,000	0,000	0,000
	Best Iteraction	177,280	33,035	135,000	152,960	19,938	120,000
Rosenbrock	Cost	0,000	0,000	0,000	0,000	0,000	0,000
	Best Iteraction	2469,840	355,367	1347,000	26637,920	3776,423	22680,000
Schwefeld	Cost	-2066,489	51,626	-2094,914	-2094,914	0,000	-2094,914
	Best Iteraction	1645,920	2928,706	133,000	193,080	49,071	129,000

Table 4: $c_1 = 0.5$, $c_2 = 2.0$, $c_3 = 2.0$

The preceding tables allow comparing the original PSO algorithm to the proposed PSO-T. This will be done in the next section.

4. Final remarks

This work presented a new version of PSO algorithm, denoted as PSO-T (*Particle Swarm Optimization with Turbulence*). These new version introduces a turbulent term that affects in an independent, random and sporadic way the flight of each bird of the flock. Minimization tests of real functions were performed comparing the PSO-T to the PSO and allow to conclude that the turbulence is effective to escape from local minima in the search space of these functions, yielding better solutions. For example, in Table 2, function Griewank, PSO stops in a local minima after 173,96 iterations (in mean), while PSO-T, stops in a local minima only after 29801,36 iterations (in mean) and gets a better value for the function (0,10 against 0,524 of canonical PSO), with a lower standard deviation. The same comments can be applied to Rastrigin and Rosenbrock functions. In addition, the turbulence gives more robustness to the algorithm, since the side effects of choosing a far from optimum set of parameters (c_1, c_2 and c_3) is diminished. This is due to the higher capability of perform a random search that is provided by the turbulence. This behavior is clearly seen in Tables 3 and 4, where at least one seed of PSO-T (but none of PSO) finds the global optima. In [1] PSOT is applied to Thermal-Vacuum Modelling, to find out fuzzy models to represent dynamic behavior of space systems that lie underneath the space qualification process. It is intended to employ the PSO-T for solving inverse problems, which are usually iteratively solved as optimization problems. For instance, another metaheuristic, the Ant Colony Optimization, was successfully applied to such problems as can be seen in [7].

References

[1] E. Araujo, F. P. A. Araujo, J. C. Becceneri, H. F. de Campos Velho (2009). “Particle Swarm Optimization with Turbulence (PSOT) applied to Thermal-Vacuum Modelling”, Fuzzy-IEEE2009: 2009 IEEE International Conference on Fuzzy Systems, ICC Jeju, Jeju Island, Korea, August 20-24.

[2] R. Boyd and P. Richarson, “Culture and the Evolutionary Process”, University of Chicago Press, 1985.

[3] M. Clerc. “The swarm and the queen: towards a deterministic and adaptive particle swarm optimization”. Proceedings of the IEEE Congress on Evolutionary Computation (CEC 1999), pp. 1951- 1957, 1999.

[4] R. C. Eberhart and J. Kennedy. „A new optimizer using swarm theory”. Proceedings of the Sixth International Symposium on Micromachime and Human Science, Nagoya, Japan, pp. 39-43, 1995.

[5] R. C. Eberhart and Y. Shi. “Particle swarm optimization: developments, applications and resources”. Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2001), Seoul, Korea, 2001.

[6] J. Kennedy. “Minds and cultures: particle swarm implications. Socially Intelligent Agents”. Paper from the 1997 AAAI Fall Symposium, Menlo Park, CA, pp. 67-72, 1997.

[7] R. P. Souto, S. Stephany, J. C. Becceneri, H. F. C. Velho, A. J. Silva Neto. “On the use of the ant colony system for radiative properties estimation”. Proceedings of the 5th International Conference on Inverse Problems in Engineering - Theory and Practice (ICIPE). Leeds, Inglaterra: Leeds University Press, pp. 1-10, 2005.

[7] R. B. Stull. “An Introduction to Boundary Layer Meteorology”, Kluwer Academic Publishers, 1988.