COSO: Community Of Scientists Optimization

Alfredo Milani[1], Valentino Santucci[1]

[1] University of Perugia, Department of Mathematics and Computer Science,
via Vanvitelli 1, 06100 Perugia, Italy
{milani, valentino.santucci}@dmi.unipg.it

Abstract. In this paper we propose COSO "Community Of Scientists Optimization": a new evolutionary and population-based optimization algorithm using, as metaphor, the behavior of the research world, that is, probably, the most notable example of swarm intelligence. It is built as a variant of Particle Swarm Optimization (PSO) and the main idea introduced is, fundamentally, the use of a budget value associated to each population's individual. Basing on these values, COSO is able to evolve the population size, generating and killing entities, other than the properties of the individuals. Experiments shown that COSO generally improves the convergence speed with respect to basic PSO.

Keywords: Evolutionary algorithms, swarm intelligence, particle swarm optimization.

1 Introduction

Optimization problems can be defined as the computational problems in which the object is to find the best of all possible solutions. More formally, find a solution in the feasible region which has the minimum (or maximum) value of the given objective function. Indeed, the problem of minimizing the function $f: \Theta \rightarrow \Re$ with $\Theta \subseteq \Re^n$ (the region of feasible solutions) can be stated as finding the value $\theta^* = arg\ min\, f(\theta)$ with $\theta \in \Theta$.

This kind of problems are frequently encountered in various fields like computer science, mathematics, economy, computational chemistry, etc. Generally the instances of these problems present a very large solutions' space and, because of it, the most important instances belong to the class of NP problems for which a polynomial and exact algorithm does not exist. For this reason a lot of approximated and stochastic optimization schemas are been introduced; recently the algorithms that are gain more attention are the evolutionary and population-based algorithms [5] that are able to locate the optimal (or a near-optimal) solution more quickly than others, especially in the optimization of non-linear functions.

Evolutionary algorithms are iterative methods using a given number of entities, interacting with each other, that navigate the search space in order to find the minimum (or maximum). The entities represent the candidate solutions to the problem and, at each iteration, they are modified in some way based on their fitness values.

A lot of evolutionary algorithms are been developed; the more famous are Genetic Algorithms (GA) [4], Particle Swarm Optimization (PSO) [1] and Ant Colony Optimization (ACO) [6]. As the names suggest they are based on some natural mechanisms; for example, PSO takes inspiration from the animal social behavior, such as fish schooling or birds flocking.

The evolutionary algorithm that we present and whose name is COSO "Community Of Researchers Optimization" is a PSO's variant and takes origin from the following observations:

- "*It's a good idea to take inspiration from the behavior of some kind of animals (birds, fish, ants, ...) for developing an evolutionary optimization method, but why exclude just that one we believe the best,i.e. the man?*"
- "*Why exclude just the distributed process that produce the highest results in the advancing of knowledge, i.e. the method used by the scientific community?*"

Indeed, the research world is probably the most notable example of swarm intelligence. Each member of the researchers' community, interacting with the others, produces new papers and hence new ideas that increment the amount of human's knowledge.

Basing on these considerations COSO introduces some new aspects in the original PSO schema. One of these is the use of a dynamic population size of researchers/particles, but the most original idea introduced in this paper, at the best of our knowledge, is the research fund (or budget value) associated to each individual of the swarm.

After recalling the algorithmic fundamentals of PSO, we introduce COSO in paragraph 3 discussing different features of the model. In paragraph 4 we analyze the new parameters introduced in COSO and, finally, experiments and results are presented and discussed in paragraph 5.

2 Particle Swarm Optimization

PSO has been introduced by Kennedy and Eberhart [1] and got its inspiration from particles' model of objects and simulation of collective behavior of flocks of birds.

A PSO's swarm is composed of a set of particles $P = \{p_1, p_2, \ldots, p_n\}$ interconnected in a graph that define a neighborhood relation among particles, i.e. for each particle p_i is defined its set of neighbors $N_i \subseteq P$. The position of a particle represents a candidate solution of the given optimization problem represented by an objective/fitness function $f: \Theta \rightarrow \Re$ with $\Theta \subseteq \Re^n$ (the region of feasible solutions) to be minimized (maximized). Every particles, as time passes, through its search adjusts its position according to its own experience as well as the experience of its neighbors. In this way PSO combines cognitive and social strategies in order to focus the search of the swarm toward the most promising areas.

At any step t, every particle p_i is associated to the following n-dimensional vectors (n is the dimensionality of the search space):

- $x_{i,t}$, i.e. the position,
- $v_{i,t}$, i.e. the velocity,

- $b_{i,t}$, i.e. the particle personal best that is the best position that p_i has ever visited until time step t,
- $l_{i,t}$, i.e. the best position ever found among its neighbors until time step t.

In the more standard PSO implementation a complete network is used as neighborhood graph. In this case the $l_{i,t}$ values are substituted by a global l_t that is the same for all the particles and can be maintained in a more efficient way.

PSO starts by generating random positions within the feasible region Θ, moreover velocities are usually initialized to small random values to prevent particles from leaving the search space in the first iteration.

During the main loop velocities and positions are iteratively updated until a stop criterion is met. The update rules are:

$$x_{i,t+1} = x_{i,t} + v_{i,t+1} \quad \textbf{(1)}$$

$$v_{i,t+1} = \omega\, v_{i,t} + \varphi_1\, \beta_{1,t}\, (b_{i,t} - x_{i,t}) + \varphi_2\, \beta_{2,t}\, (l_{i,t} - x_{i,t}) \quad \textbf{(2)}$$

Weights in equation (2) respectively represent the inertia ω, the acceleration factors φ_1, φ_2 and the random factors $\beta_{1,t}$, $\beta_{2,t}$ which are uniformly distributed in [0,1]. The three terms in the velocity-update rule (2) characterize the behavior of the particles. The first term, called the inertia or momentum, serves as a memory of the previous flight direction and prevents a particle from drastically change its direction. The second term, called the cognitive component, models the tendency of the particles to return to the previously found best position. The third term, i.e. the social component, quantifies the velocity's contribution relative to neighbors.

The best positions (personal and social) are updated, as obvious, following these rules:

$$b_{i,t+1} = x_{i,t+1} \text{ if } f(x_{i,t+1}) < f(b_{i,t}) \quad \textbf{(3)}$$

$$l_{i,t+1} = min\ b_{j,t+1} \text{ with } j \in N_i \quad \textbf{(4)}$$

In the most common case of a topology formed by a fully connected graph equation (4) becomes:

$$l_{t+1} = min\ b_{i,t+1}\ \forall i \quad \textbf{(5)}$$

Finally, note that sometimes the particles can go outside the bounds of the feasible search space Θ. In our work, in order to solve this problem, we randomly restart the position of the out-of-bounds' particles.

3 Community Of Scientists Optimization (COSO)

COSO is a PSO's variant using the research activity as metaphor. The research world is a dynamic community where the members, i.e. scientists or researchers, compete and cooperate for publish their papers in a journal, i.e. researchers add their ideas to a knowledge gathering point.

Following this analogy COSO is composed by a dynamic set of researchers $P = \{p_1, p_2, ..., p_n\}$, i.e. the PSO's particles, that share a journal J, i.e. the communication channel, and compete for publish their papers, i.e. the researchers' positions, in that journal.

Each researcher is associated to the same properties of PSO's particles other than to a non-negative integer value $m_{i,t}$ representing the amount of research fund (or moneys) assigned to the i-th researcher at time t. When a researcher starts its activity m_i is initialized to a given value m_0. Apart from this, the initialization phase is the same as in PSO.

In every generation each researcher submits a paper to the journal, i.e. he evaluates the fitness function in its current position $x_{i,t}$. As in standard PSO during a generation the researchers' velocities and positions are updated, moreover in COSO an update rule for the money values and a mechanism that modifies the population size must be defined. In this way the population size can vary. In the following we denote with n_t the number of researchers working at generation t; finally, note that the population size is initialized to a given value n_0.

3.1 The Journal

The journal J_t is an ordered set of the best positions ever visited until time t by all the researchers. J_t plays the role of the value l_t used in PSO. In this way COSO can consider the best k positions of ever and not only the best one position; metaphorically, it means that only the best papers are published in the journal. Moreover k is called the journal length.

According to this variant the velocity update equation (2) is modified as follow:

$$v_{i,t+1} = \omega v_{i,t} + \varphi_1 \beta_{1,t} (b_{i,t} - x_{i,t}) + \varphi_2 \beta_{2,t} (J_{t,r} - x_{i,t}) \quad \textbf{(6)}$$

where $J_{t,r}$ is a component of the journal with r randomly chosen in $[1,k]$. Note that the rest of the equation is the same as in PSO. Using not only the global best position of ever in the social component prevents in part the convergence to local optima.

Finally, the position update equation (1) remains unmodified.

3.2 The Moneys

Since the main innovation of COSO is the budget value associated to individuals, another rule needs to be added to the framework in order to update the researcher's money value $m_{i,t}$.

After the process of fitness evaluations the amount of moneys per researcher is decreased of γ, meaning that the researcher has spent a part of his fund for the paper submission activity. Hence:

$$m'_{i,t+1} = m_{i,t} - \gamma \quad \textbf{(7)}$$

From (7) it's clear that the total amount of moneys spent in each paper submission phase is exactly γn_t. Now, the idea is to redistribute this quantity to reward the authors of the best papers in the last generation according to a given distribution function d. As obvious, d can assume different forms; in this work we propose a function that maintains constant the total amount of moneys during the generations and rewards majorly the best researcher with $\lfloor \lambda \gamma n_t \rfloor$ moneys and assign μ moneys to the others in the ranking list of the last generation (note that if μ is not 1 the last rewarded can receive less than μ moneys). Given that choice, the number of rewarded researchers is $1+\lfloor \lceil (1-\lambda)\gamma n_t \rceil / \mu \rfloor$ and the distribution function d can be written as follow:

$$d(m_{i,t+1}) = \begin{cases} m'_{i,t+1} + \lfloor \lambda \gamma n_t \rfloor & \text{if } p_i \text{ is the } 1^{st} \\ m'_{i,t+1} + \mu & \text{otherwhise} \end{cases} \quad \textbf{(8)}$$

In our settings the parameter λ represents the percentage of the total jackpot assigned to the best, μ is the amount of moneys won by the other researchers in the ranking and γ represents the quantity of moneys spent at every fitness evaluation. The values used in the experiments are respectively 0.5, 1 and 1.

3.3 Population Dynamics

COSO's population evolves basing on the researchers' budget values. At the end of each generation the richest researchers are able to reproduce and generate new offspring while the poorest die.

In our model a reproduction threshold ρ is established and the number of offspring generable from each researcher is determined by ρ, m_0 and $m_{i,t}$.

The researchers who have no more money, i.e. $m_{i,t} = 0$, die and finish its research activity. Instead the researchers that have reached or exceeded the reproduction threshold, i.e. $m_{i,t} \geq \rho$, generate new offspring as much as possible with the constraints that each offspring j must had an initial amount of moneys equal to m_0, i.e. $m_{j,t} = m_0$, and the parent researcher must retain at least m_0 moneys.

In this way, the researchers that have did a good work in their recent research activity are rewarded, while the others are penalized.

3.4 Offspring Initialization

A new researcher j, generated by a parent researcher h, inherits the position and best personal property from the father, i.e. $x_{j,t}=x_{h,t}$ and $b_{j,t}=b_{h,t}$.

Instead the offspring's velocity is initialized according to

$$v_{j,t+1} = \omega\, v_{h,t} - \varphi_1\, \beta_{1,t}\, (b_{h,t} - x_{h,t}) - \varphi_2\, \beta_{2,t}\, (J_{t,r} - x_{h,t}) \quad \textbf{(9)}$$

where the variables and the parameters are the same of equation (6) but this time the cognitive and the social components (second and third terms) are subtracted and not added. This can be consider as a repulsion [7] and it is applied in order to repel the new individuals from the swarm and promote the exploration of undiscovered areas of the search space. On the other hand the inherited properties guarantee that, in the case of an unsuccessful research, the offspring will be attracted back to parent's trajectory during the upcoming generations.

4 Parameters

Basic PSO has a small number of parameters that need to be fixed. One of these is the population size *n* that is often set empirically basing on the dimensionality and the perceived difficulty of the problem; however, values in the range 20-50 are quite common [2]. The other parameters are those appearing in equation (2), i.e. the inertia weight ω and the acceleration factors φ_1 and φ_2. It is demonstrated in [3] that if the values of ω, φ_1 and φ_2 are properly chosen, it is guaranteed that the particles' velocities do not grow to infinity. In [2] the suggested values for the parameters are respectively 0.7, 1.43, 1.43 and these are used in our experiments.

In COSO other parameters are introduced; they are:

- the journal length k,
- the initial amount of moneys m_0,
- the reproduction threshold ρ,
- the prizes distribution function d.

Obviously, d can assume different forms; in this work we propose a function that maintains constant the total amount of moneys during the generations like the one of equation (8). Indeed, this function redistributes as prize the same number of moneys spent by the researchers in the last function evaluation session.

Fixed the prizes' distribution function d, the parameters m_0 and ρ influence the variation of the population size n_t. Obviously the reproduction threshold of a researcher must be greater than the initial amount of moneys but, more strongly, must hold the following relation:

$$\rho \geq 2m_0 \qquad \textbf{(10)}$$

Indeed, if (10) does not hold, a researcher is not able to reproduce. From ρ depends also the number of generations to attend before reproducing. Experimentally we have deducted that the best choose is $2m_0 \leq \rho < 3m_0$; in this way, when the reproduction threshold is reached from a certain researcher, except for the first of the last ranking that receive a big prize, only one offspring is generated and the parent retains m_0, or something more, moneys.

The initial moneys parameter m_0 and the initial population size n_0 give the total budget divided among the researchers at each iteration. The value of m_0, other than respects equation (10), must be chosen small in order to decrease the amount of "bad" papers needed for removing a researcher from the swarm.

The journal length influences in some way the degree of exploration of COSO and if chosen great is able to decrease the probability of convergence to a local optima; however k influences also the complexity of the paper publication phase; a good compromise that we have saw experimentally is to chose $k=\lfloor n_0/2 \rfloor$.

Finally, note that the two main difference between COSO and basic PSO are: the presence of a journal instead of a single global best and the reproduction mechanism that changes the population size. For these reasons we can think at COSO as a basic PSO's generalization, indeed, if we choose k=1, $\rho<2m_0$ and m_0 sufficiently large (for example it can be set to the maximum number of generations allowed) the velocity update equation (6) becomes the (2) and the population size remains constant during the generations excluding hence the reproduction phase, i.e. the basic PSO's behavior is emulated.

5 Experimental Results

The performances of COSO have been evaluated on 5 benchmark functions which differ for the properties of modality, separability and symmetry around the global optimum. The used test bed functions have been suggested in [11] and are described in Table 1, where the number of dimension (*D*) of the search space, the search interval (*Int.*) and the properties of each function are given.

Table 1. Test bed functions.

f	Name	D	Int.	Mod.	Sep.	Sym.
f_1	Sphere	10	[-100,100]	uni	yes	yes
f_2	Rosenbrock	10	[-30,30]	uni	no	no
f_3	Ackley	10	[-30,30]	mul	yes	yes
f_4	Rastrigin	5	[-5.12,5.12]	mul	yes	yes
f_5	Griewank	5	[-600,600]	mul	no	yes

For each experiment the population size is set to 20 for f_1, f_2, f_3 and to 40 for f_4, f_5; the algorithm has been run for 100 times and the averaged fitness is reported.

A first session of experiments have been held in order to test the effect of different chooses of the parameters' couple ρ and m_0.

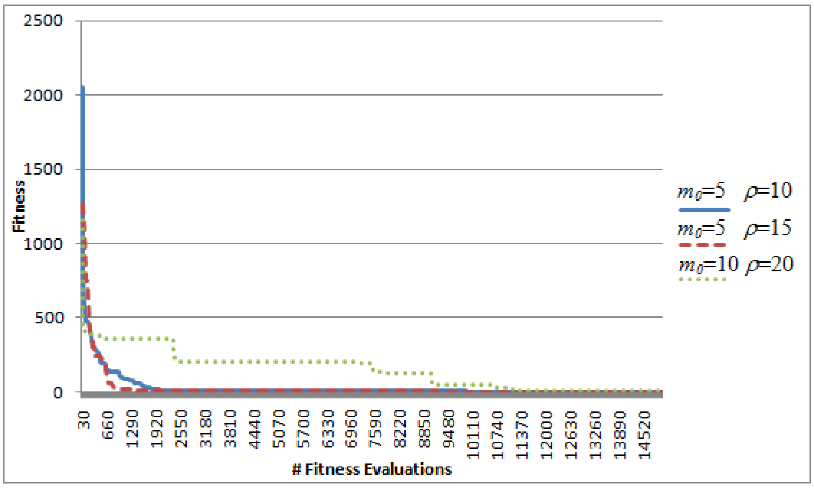

Fig. 1. Parameters' comparison for f_2.

Basing on the results of these experiments, reported in the convergence graph of fig. 1, the parameters m_0 = 5 and ρ = 10 are chosen for the comparison's experiments with basic PSO. The comparison is summarized in table 2 where, both for COSO and PSO, the number of fitness evaluations (NFE), needed to converge at the optimum fitness value, is reported. As convergence criteria we have used an error value of 1e-6 within the optimum. Note that other than the averaged value is reported also the best convergence value. Each test is run for a maximum of 2e6 NFE.

Table 2. COSO-PSO comparison.

f	COSO		PSO	
	Avg	Best Run	Avg	Best Run
f_1	7866	6069	10020	6160
f_2	434687	354780	>2000000	>2000000
f_3	22199	18733	>2000000	19620
f_4	>2000000	10451	>2000000	>2000000
f_5	800043	3368	>2000000	30345

From the experiments results that COSO better performs basic PSO, indeed, generally, COSO converges faster than PSO. In the case f_4, COSO, like PSO, is not able to reach the optimum (before the maximum NFE allowed) in the averaged measure; however, the best run measures obtained mean that COSO certainly has a convergence's frequency greater than those of PSO.

Finally, note that, in our implementation, the "Mersenne Twister" [10] algorithm is used as method for generating uniformly-distributed random numbers.

6 Conclusion and Future Works

In this paper a variant of PSO, using the research world as metaphor, has been introduced. The main innovation is the use of a budget value, associated to each individual, that allow new dynamics in the population's evolution. Indeed, COSO introduces a reproduction mechanism that is able to dynamically change the number of working entities.

Experiments have been held on some common test bed functions and they shown that COSO generally improves the convergence speed with respect to basic PSO. However, further experiments are needed to confirm the positive results for a wider range of functions and also for comparing COSO with other dynamic PSOs like those presented in [8] and [9].

Finally, another future work is the study of different use of the budget values, for example taking inspiration from economy or game theory.

References

1. J. Kennedy, R. Eberhart. “Particle swarm optimization”. In Proc. of IEEE Conf. on Neural Networks, IEEE Press, 1995, pp. 1942-1948
2. R. Poli, J. Kennedy, T. Blackwell. “Particle swarm optimization. An overview.”. Swarm Intelligence, 1(1): 33-57, 2007
3. M. Clerc and J. Kennedy. The particle swarm-explosion, stability and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation, 6(1):58-73, 2002
4. Holland, John H (1975), Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor
5. Bäck, T. (1996) Evolutionary Algorithms in Theory and Practice, Oxford, NY
6. M. Dorigo, V. Maniezzo, and A. Colorni. Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics – Part B, 26(1):29–41, 1996
7. Jacques Riget and Jakob S. Vesterstrøm. A Diversity-Guided Particle Swarm Optimizer – the ARPSO. EVALife Technical Report no. 2002-02
8. Yanjiang Miao, Zhihua Cui, Jianchao Zeng, "Dynamic Population-Based Particle Swarm Optimization Combined with Crossover Operator," his, vol. 1, pp.399-404, 2009 Ninth International Conference on Hybrid Intelligent Systems, 2009
9. Qi Kang, Lei Wang and Qi-di Wu. A novel ecological particle swarm optimization algorithm and its population dynamics analysi. Applied Mathematics and Computation Volume 205, Issue 1, 1 November 2008, Pages 61-72
10. Makoto Matsumodo, Takuji Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation (TOMACS), Volume 8 , Issue 1 (January 1998)
11. Price, K V, R M Storn, e J A Lampinen. Differential evolution: a practical approach to global optimization. Springer Verlag, 2005.