# Balancing Exploration and Exploitation in Nature Inspired Computing Algorithm

**K. Praveen Kumar, Sangeetha Singarapu, Mounika Singarapu, and Swaroop Rakesh Karra**

**Abstract** Nature is the one of the best inspiration for solving problems around us. Nature inspired computing algorithms are inspired from nature. These algorithms have inbuilt features of self learning, self motivation, co-ordination and collective behavior for solving a particular task. These nature inspired computing algorithms are specified by local search (exploitation) and global search (exploration). As per the effects of nature inspired computing algorithms, depends the searching capability. This paper focuses balance of local and global searching attribute of nature inspired computing algorithms. The people in a group are managed by leaders. By applying PSO algorithm, the best leader in the group is selected using efficient fitness functions. And this proposed fitness function is compared with the benchmark fitness function for providing better optimum results in the balancing of exploration and exploitation.

**Keywords** Nature inspired computing algorithms · Swarm intelligence · Particle swarm optimization · Local search · Global search

K. P. Kumar (✉) · S. Singarapu
Department of IT, KITSW, Warangal, India
e-mail: kpk.it@kitsw.ac.in

S. Singarapu
e-mail: ss.it@kitsw.ac.in

M. Singarapu
Dept. of MS ITM, Campbellsville University, Louisville, KY, USA

S. R. Karra
MS IT, New Castle Wilmington University, New Castle, DE, USA

J. Hemanth et al. (eds.), *Intelligent Cyber Physical Systems and Internet of Things*,
Engineering Cyber-Physical Systems and Critical Infrastructures 3,
https://doi.org/10.1007/978-3-031-18497-0_12

# 1 Introduction

Nature is the great inspiration for solving complex problems in the universe. Nature İnspired Computing (NIC) algorithms are inspired from nature for solving complex problems and produce nearest results at reasonable amount of time [1, 2]. The NIC algorithms having features of collective behaviour, self learning, self adapt nature are focused to solve NP hard problems. The nature inspired computing algorithms have feature of random probability i.e., algorithms randomly pick solutions for better optimum results [3–6].

Particle Swarm Optimization (PSO) is a popular nature inspired computing algorithm proposed by Dr. Eberhart and Dr. Kennedy in 1995. PSO is influenced by social behavior of birds and animals [7–10]. PSO is a well known nature inspired computing algorithm for the optimization of initializing random population of solutions. PSO updates component generations for searching best positions. Their particles manage their velocity of own positions and also positions of their neighbor particles to get optimum solutions.

# 2 Existing Work

Paper [11] focused on particle swarm optimization about local search (exploitation) and global search (exploration) by applying master and slave method. The leader swarm acts, master and other swarms act and then the slaves. The slave swarms provide new positions along best fitness values to master swarms. The position of new particles is updated by master swarm for best position and manage updated positions of themselves (master swarms) and update slave swarms' position.

Article [12] proposed reduced energy conservation model of power system for socio-economic development by applying modified particle swarm optimization. Balancing of local search as well as global search using a new technique of multi-swarm was proposed. In the article, 'clan' was introduced in place of swarms, and every group had a leader. Representing local best is the best solution that is represented by the leader of clan [13, 14]. To select global best, all these leaders can represent themselves for selection procedure. The clans update their positions, and manage their population risks for selection of local and best positions accordingly.

## 2.1 *Particle Swarm Optimization*

PSO is initialized with randomly generated population of particles (initial swarm) and a random velocity is assigned to each particle that propagates the particle in search space toward optima over a number of iterations. Each particle has a memory, remembering best position attained by it in the past, which is called personal best position (Pbest). Each particle has its Pbest and the particle with the best value of fitness is called global best particle (Gbest). Suppose that the search space is D dimensional, the ith particle of the population can be represented by a D-dimensional vector (xi1, xi2, …, xiD)T. The velocity of this particle can be represented by another D-dimensional vector (Vi1, Vi2, …, ViD)T.

The previously best visited position of ith particle is denoted by Pi and the best particle in the swarm is denoted by Pg. The update of the particle's position is accomplished by the following two equations. Eq. (1) calculates a new velocity for each particle based on its previous velocity and Eq. (2) updates each particle's position in search space.

$$V_{id}^{k+1} = wV_{id}^{k} + c_1 r_1 \left[ p_{id}^{k}(t) - x_{id}(t) \right] + c_2 r_2 \left[ p_{g}^{k}(t) - x_{id}^{k}(t) \right] \tag{1}$$

$$x_{id}^{k+1}(t+1) = x_{id}^{k}(t) + v_{id}^{k+1}(t+1) \tag{2}$$

where k = iteration number, d = 1,2,3,…, D, i = 1,2,3,…, N, N = swarm size, w = inertia weight which controls the momentum of particle by weighing the contribution of previous velocity, c1 and c2 are positive constants called acceleration coefficients and r1 and r2 are random numbers uniformly distributed between [0,1].

PSO particle represents D-dimensional area as particle i (particle i1, particle i2, particle i3… particle iD), where D represents dimension and particle iD minimum dimension, maximum dimension is mini and max limitations of Dth dimension. Velocity of particle i is velocity i = (velocity i1, velocity i2, velocity i3…. Velocity iD). Particles and time t made manipulations as per the following relations.

$$\begin{aligned} &Velocity\,i\,(time\,t+1) = velocity\,i\,(time\,t)\\ &+r\,value\,1\,con\,1\,(Ppbp\, -\, particle\,i\,(time\,t))\\ &+r\,value\,2\,con\,2\,(Pgbp\, -\, particle\,i\,(time\,t)) \end{aligned} \tag{3}$$

$$Particle\,i\,(time\,t+1) = particle\,i\,(time\,t) + velocity\,i\,(time\,t) \tag{4}$$

In above equation, r value1 and r value2 are random values in the range 0 and 1. The con 1 and con 2 are the velocity constants that improve the movement of particles for a given iteration. The last best position of ith particle is noted as Ppbp and the best position of ith· particle is noted as Pgbp.

$$\begin{aligned} &Velocity\,i\,(time\,t+1) = inertia\,w\,x\,velocity\,i\,(time\,t)\\ &+rvalue1con1\,(Ppbp\, -\, particle\,i\,(time\,t))\\ &= rvalue2con2\,(Pgbp\, -\, particle\,i\,(time\,t)) \end{aligned} \tag{5}$$

In the above equation, inertia is added [15]. For a perfect balancing between the global and local explorations through inertia, minimum iterations are considered to achieve optimal solution.

$$inertia = inertia\,max = \frac{inertia\,max - inertia\,min}{iterations\,max} \times iterations \tag{6}$$

In above equation, inertiamax is denoted by initial weight, inertiamin is denoted by final weights, iterationsmax is max number of iterations considered in each iteration which is denoted as the number of iterations at present.

In addition, Linearly decreasing inertia weight method in PSO (LPSO) is a random inertia weight factor for dynamic systems tracking [16]. Inertia weight factor noted for random vary as per the following equation.

$$inertia\; = 0.5 - \frac{rand()}{2} \tag{7}$$

In above equation, rand() denotes equally allocated random numbers in between values 0 to 1 range. Value 1.494 must be maintained by acceleration coefficients. This equation refers to Random weight method in PSO (RPSO).

## 3 Proposed Work

### *3.1 Improved Multiple Particle Swarm Optimization (IMPSO)*

The main objective of Extensible Multiple Swarm Particle Swarm Optimization is the co-ordination behaviour of communication among the people in the group ‘clan’ (group of people) as in Fig. 1. The leaders of the groups search for optimum solutions. The swarm (group birds) is called as clan (group of people), and every clan (group of members) have solutions produced by the members of group. The best member of every clan is treated as the leader of that clan. A new leader of the clan is selected every generation due to the changing of positions of the members for meeting participation. The inertia function control exploration (global best) presents the algorithm. The main task of clan is to manage and control other members of the clan for movement and time control. Based on the new available position, every leader meet at one room (location) to find the best leader.

**Fig. 1** Swarm optimization of clan with individual members

PSO algorithm is used for balancing the position for exploration as well as to find exploitation. Based on PSO, the Optimization of Meeting Room Method (OMRM) is performed. İn this, every member in the clan is equal to every particle in swarm. The position as well as velocity is updated by PSO. The proposed algorithm's stepwise performance, at first step shows to select the best leader (in a group of clan) who is treated as local best. In second step, overall best leader (in clan groups) is selected and is treated as global best. The optimization is determined by OMRM in the meeting room as in Fig. 2. By executing this algorithm, those who are selected as the overall best leaders, share their position with other leaders.

$$iWeight^{nl} + \left(\frac{iWeight^{gl} - iWeight^{nl}}{iteration_{\max}}\right) x\, rand() = \tag{8}$$

$$\begin{aligned}(time_t + 1) &= iweight^{nl}\, x\, Velocity_i^{nl}(t)\\ &+ rvaluecon(Particle_{global}^{Leader} - Particle_{local}^{Leader}(time_t))\end{aligned} \tag{9}$$

$$particle_i^{nl}(\mathrm{time_t} + 1) = particle_i^{nl}(\mathrm{time_t} + 1) + Velocity_i^{nl}(time_t) \tag{10}$$

In above equation, nl denotes the normal_leaders, gl represents the overall global leader, partile i states the position of the normal_leaders, $Velocity_i^{nl}$ states the velocity of normal leaders, $iWeight^{gl}$ and $iWeight^{nl}$ denote inertia weight of global leader and normal leader respectively.

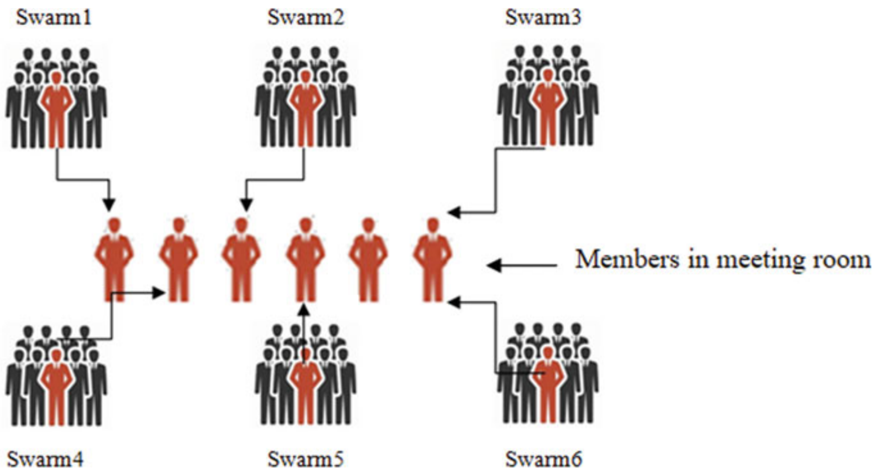

**Fig. 2** Meeting room approach

**Algorithm for Improved Multi-Particle Swarm Optimization**

See Fig. 3**.**

## 4 Results and Discusssion

The performance of Improved Multiple Particle Swarm Optimization (IMPSO) is evaluated using benchmark functions compared with MPSO, Master–Slave SPO (MSPO) [11], and Multi-swarm Cooperative Particle Swarm optimizer (MCPSO) [17]. Parameters used are same for all these algorithms. The number of iterations used is 500 with 50 dimensions. For every experiment, 30 runs are performed. Testing with function is shown in Table 1 and parameter values in Table 2.

Table 3 indicates the performance optimum results of particles for benchmark functions. EMPSO perform best results when compare to other algorithms. IMPSO has 6 swarms, every group consists of 10 particles and 6 particles in the room. IMPSO has less computation and shows better performance for optimum solutions.

In Fig. 4a, by applying Sphere function, simulation results of IMPSO shows better performance when compared with other algorithms.

In Fig. 4b, by applying Griewank function, simulation results of IMPSO shows better performance than that of MCPSOS, SPSO, MPSO, IMPSO [12].

```
Input
#swarm, $\#particle_i$, con1, con2,#dimension,#maxgen
Output
Best solution(leader)
Procedure
Start
    Swarm initialization
    For each particle evaluate fitness
    while(iteration≤ #maxgen)
        For every s in swarm
            for every swarm member s
             For updation use equation3 for velocity of member in swarm
                  For updation use equation2 for position of member in swarm
            next
            select local best as leaders
Next
Among all leaders select the best leader
For updation use equation6 for inertia weight
For updation use equation7 for leaders velocity
For updattion use equation8 for leaders position
Selection of Best_leader for global_best
Loop
Return to best_leader
Stop
```

**Fig. 3** Improved multi particle swarm optimization algorithm

**Table 1** Testing with function

| Function name | Function |
|---|---|
| Sphere unimodal | $\mathrm{f}_1 = \sum_{i=1}^{D} X^2$ |
| Griewank unimodal | $\mathrm{f}_2 = \sum_{i-1}^{D} -\mathrm{x}_{1/4000}^2 - \prod_{i=1}^{D} \cos \frac{\mathrm{x_i}}{\sqrt{\mathrm{i}}}$ |

## 5 Conclusion

The paper shows the importance of Swarm Intelligence algorithms for solving real-time problems. Swarm Intelligence algorithms are related to Nature Inspired Computing algorithms. The social group behavior of human being is proposed by applying swarm intelligence algorithm. The proposed algorithm Improved Multi-purpose particle swarm optimization manages the local search (exploitation) as well as the global search (exploration) techniques. By applying benchmark functions, the results of the extensible multi purpose swarm optimization shows better performance

**Table 2** Parameter values

| Algorithm | Parameter | Value |
|---|---|---|
| SPSO | W (inertial weight) | 0.9–0.4 |
| | Number of swarms | 1 |
| | Swarm size (population) | 50 |
| | c1, c2 (velocity constants) | 1.5 |
| MCPSO | W (inertial weight) | 0.9–0.4 |
| | Number of swarms | 1 |
| | Swarm size | 50 |
| | c1, c2(velocity constants) | 1.5 |
| MSPO | Wln | 0.8–0.5 |
| | Wlg | 0.9–0.7 |
| | c1,c2(velocity constants) | 1.5 |
| | Number of clans | 5 |
| | Clan size | 10 |
| IMPSO | Inertianl | 0.8–0.9 |
| | Inertiagl | 0.9–0.9 |
| | Constant1, constant2 | 1.5 |
| | Number of clans | 6 |
| | Clan size | 10 |

**Table 3** Benchmark test functions' results

| Fn | Swarm | Algorithm | Best |
|---|---|---|---|
| F1 | 50 | SPSO | 2.5457521 |
| | | MCPSO | 0.9854126 |
| | | MPSO | 0.0007845 |
| | | EMPSO | 0.0006555 |
| F2 | 50 | SPSO | 0.0884741 |
| | | MCPSO | 0.0078414 |
| | | MPSO | 0.0000897 |
| | | IMPSO | 0.0000500 |

than the existing swarm optimization algorithms. The future work plans to implement hybrid nature inspired computing algorithms for optimum results.

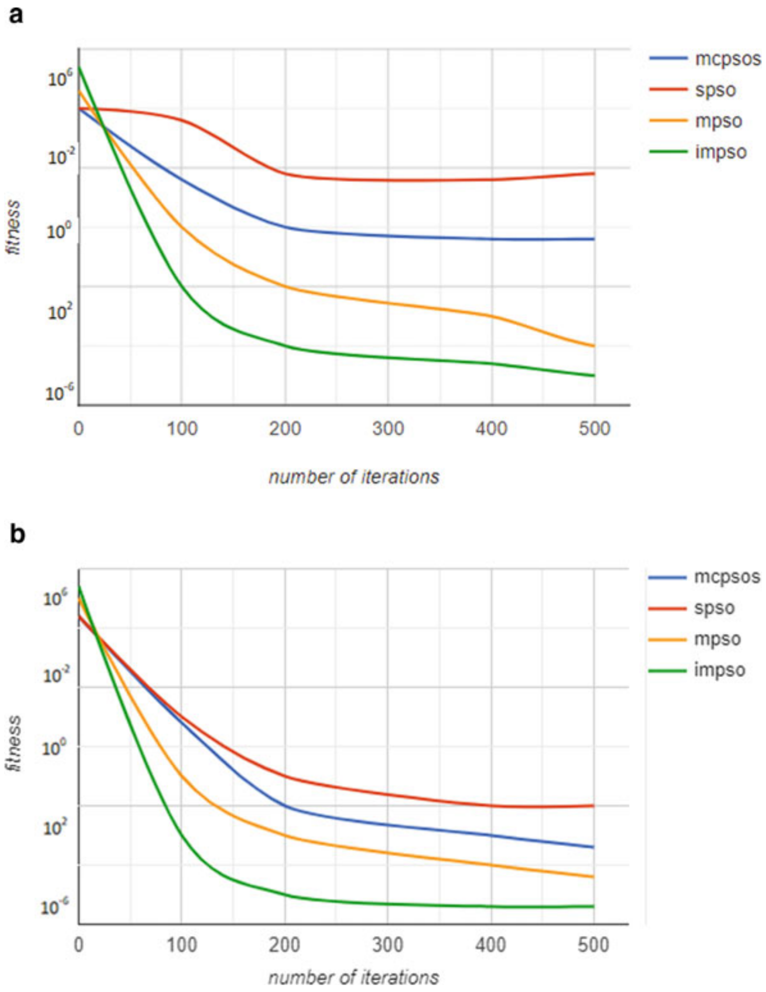

**Fig. 4** **a** Convergence curve using sphere function, **b** convergence curve using Griewank unimodal function

## References

1. Binitha S, Sathya SS (2012) A survey of bio inspired optimization algorithms. Int J Soft comput Eng 2(2):137–151
2. Dixit M, Upadhyay N, Silakari S (2015) An exhaustive survey on nature inspired optimization algorithms. Int J Softw Eng Appl 9(4):91–104
3. Slowik A, Kwasnicka H (2018) Nature inspired methods and their industry applications—swarm intelligence algorithms. IEEE Trans Ind Inform 14:1004–1015
4. Diao R, Shen Q (2015) Nature inspired feature selection meta-heuristics. Artif Intell Rev 44:311–340

5. Azrag MAK, Kadir TAA, Odili JB, Essam MHA (2017) A global African Buffalo optimization. Int J Softw Eng Comput Syst 3:138–145
6. Odili JB, Kahar MNM, Anwar S (2015) African Buffalo optimization: a swarm-intelligence technique. Procedia Comput Sci 76:443–448
7. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the 6th ınternational symposium on micro machine and human science 1995, MHS 1995, pp 39–43
8. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE ınternational conference on neural networks 1995, vol 4, pp 1942–1948
9. Zhang Y, Wang S, Ji G (2015) A comprehensive survey on particle swarm optimization algorithm and its applications. Math Probl Eng
10. Kameyama K (2009) Particle swarm optimization-a survey. IEICE Trans Inf Syst 92(7):1354–1361
11. Niu B, Zhu Y, He X, Wu H (2007) MCPSO: a multi-swarm cooperative particle swarm optimizer. Appl Math Comput 185:1050–1062
12. Vijayakumar T, Vinothkanna R (2020) Efficient energy load distribution model using modified particle swarm optimization algorithm. J Artif Intell 2(4):226–231
13. Korani W, Mouhoub M (2021) Review on nature-inspired algorithms. In: Operations research forum, vol 2, no 3. Springer International Publishing
14. Kumar A, Nadeem M, Banka H (2022) Nature inspired optimization algorithms: a comprehensive overview. Evolving Syst 1–16
15. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: 1998 IEEE ınternational conference on evolutionary computation proceedings. IEEE World Congress on Computational Intelligence, pp 69–73
16. Eberhart RC, Shi Y (2001) Tracking and optimizing dynamic systems with particle swarms. In: Proceedings of IEEE congress on evolutionary computation, pp 94–97. IEEE, Seoul
17. Shi Y, Eberhart RC (1999) Empirical study of particle swarm optimization. In: 1999 IEEE ınternational conference on evolutionary computation proceedings. IEEE World Congress on Computational Intelligence, vol 3, pp 1945–1950