(IJIRSE) International Journal of Innovative Research in Science & Engineering
ISSN (Online) 2347-3207

Self Accelerated Smart Particle Swarm Optimization for multimodal Optimization Problems

Anuradha L. Borkar
Electronics and Telecommunication Dept
M S S S' College of Engineering and Technology
Jalna, Maharashtra, India
anu1may@rediffmail.com

***Abstract*—This paper presents Self Accelerated Smart Particle Swarm Optimization (SASPSO) to optimize multimodal optimization problems with fast convergence. In SASPSO, the positions of particle are updated by pbest (Personal or local best) and gbest (Global best). The main advantages of SASPSO are that it doesn't require velocity equation. In addition, it does not require any additional parameter like acceleration coefficients and inertia weight as in case other PSO algorithms. Momentum factor is introduced in SASPSO which can prevent particles from out of defined region without checking the validity of positions at every iterations result in saving of computational cost. The SASPSO technique is tested with three benchmark functions and results are compared with variants of PSO. The results of benchmark functions validate that the proposed technique is more efficient for improving the quality of global optima with less computational requirement.**

Keywords: **Self Accelerated Smart Particle Swarm Optimization (SASPSO); pbest (Personal or local best) and gbest; (Global best) global best**

I. INTRODUCTION

Most of the real life problems occurring in the field of science and engineering may be modeled as nonlinear unimodal or multimodal optimization problems. Multimodal problems are generally considered more difficult to solve as there exists several local and global optima. Effective optimization of multi-dimensional and multimodal function with faster convergence and good quality of solution is a challenging task. The high computational cost and demands for improving accuracy of global optima of multimodal functions have forced the researchers to develop efficient optimization techniques in terms of new, modified or hybrid soft computing techniques

Since 1960's Genetic Algorithm (GA) has proved its dominant role in the optimization world [1-2]. The Particle Swarm Optimization (PSO) is a population-based optimization method developed by Eberhart and Kennedy in 1995 [3]. It is inspired by social behavior of bird flocking or fish schooling. It can efficiently handle problems like nonlinear, unimodal and multimodal function optimizations [4].

The new variants of PSO are proposed for faster convergence and better quality of optimum solution like Supervisor-Student Model in Particle Swarm Optimization (SSM-PSO) [5], Linear Decreasing Weight Particle Swarm Optimization (LDW-PSO) [6], Gregarious Particle Swarm Optimization (GPSO) [7], Global and Local Best Particle Swarm Optimization (GLBestPSO)[8] and Emotional Particle Swarm Optimization (EPSO)[9].

The authors propose Self Accelerated Smart Particle Swarm Optimization (SASPSO). The rest of the paper is organized in four fold. The section 2 depicts the review of original Particle Swarm Optimization (PSO). The section 3 depicts the proposed method. In section 4, experimental results on benchmark functions by proposed method and other published techniques are presented. Section 5 comprises of conclusion.

II. REVIEW OF PARTICLE SWARM OPTIMIZATION

The original framework of PSO is designed by Kennedy and Eberhart in 1995. There fore, it is known as standard PSO [3]. PSO follows the optimization process by means of personal or *local best* (p_i), *global best* (p_g), particle position or displacement (X) and particle velocity (V). For each particle, at the current time step, a record is kept for the position, velocity, and the best position found in the search space. Each particle memorizes its previous velocity and the previous best position and uses them in its movements [3]. The velocities (V) of the particles are limited in $[Vmin\ Vmax]^D$. If V *is* smaller than *Vmin* then V is set to *Vmin* or ***Xmin*.** If V greater than *Vmax* then V is set to *Vmax* or ***Xmax*.** Since the original version of PSO lacks velocity control mechanism, it has a poor ability to search at a fine grain.The two updating fundamental equations in a PSO are velocity and position equations, which are expressed as Eq. (1) and (2).

$$V_{id}(t+1) = V_{id}(t) + c_1 * r_{1d}(t) * (p_{id}(t) - X_{id}(t)) + c_2 * r_{2d}(t) * (p_{gd}(t) - X_{id}(t)) \quad (1)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (2)$$

Where,
t= Current iteration or generation.
i = Particle Number.
d= Dimensions.
$V_{id}(t)$ = Velocity of *i*-th particle for *d*-dimension at iteration *t* .
X (t) = Position of *i*-th particle for *d*-dimension at iteration *t.*
c_1 and c_2= Acceleration constants.
$r_{1d}(t)$ and $r_{2d}(t)$ = Random values [0 1] for *d*- dimension at iteration *t.*
$p_{id}(t)$= Personal or local best of *i*-th particle for *d*-dimension at iteration *t.*
$p_{gd}(t)$ = Global best for *d*-dimension at iteration *t.*

The right side of Eq. (1) consists of three parts. The first part of equation is the previous velocity of the particle. The second part is the cognition (self-knowledge) or memory, which represents that the particle is attracted by its own previous best position and moving toward to it. The third part is the social (social knowledge) or cooperation, which represents that the particle is attracted by the best position so far in population and moving towards to it. There are restrictions among these three parts and can be used to determine the major performance of the algorithm.

III. SELF ACCELARTED SMART PATICLE SWARM OPTIMIZATION

The standard PSO suffers from following disadvantages.

1. The swarm may prematurely converge when some poor particles attract the other particles or due to local optima or bad initialization
2. It has problem dependent performance. No single parameter settings exists which can be applied to all problems.
3. It requires tuning of parameters like c_1, c_2, *w*, iterations and swarm size
4. Increasing the value of inertia weight *w*, increases the speed of the particles resulting in more global search and less local search. Decreasing the inertia weight slows down the speed of the particle resulting in more local search and less global search.

So to avoid the disadvantage of standard PSO, the author proposed a Self Accelerated Smart Particle Swarm Optimization (SASPSO). In SASPSO, the positions of particle are updated by *pbest* (Personal or local best) and *gbest* (Global best particle positions) as expressed in Eq. (3). The main advantages of SASPSO are that it doesn't require velocity equation. In addition, it does not require any additional parameter like acceleration coefficients and inertia weight as the case in other PSO algorithms. Momentum factor can prevent particles from out of defined region without checking the validity of positions at every iterations result in saving of computational cost.

$$X_i(t+1)= X_i(t)+Mc*rand((X_i(t)- (gbest- pbesi_t))+(gbest-- pbesi_t)*rand) \quad (3)$$

Where,
t= Current iteration or generation.
i = Particle Number.
$X_i(t)$ = Position of *i*-th particle for *d*-dimension at iteration *t.*
r_1 and r_2 = Random values [0 1] at iteration *t.*
$pbesi_t$ = Personal or local best of *i*-th particle at iteration *t.*
Gbest = Global best at iteration *t.*
Mc= Momentum Factor

During the initial stages of the experimentation, the step size will be large and thus the positions of particles are away from the global best position. During the final stage of the experimentation, the step size is reduced to smaller value. Momentum factor can prevent particles from out of defined region without checking the validity of positions at every iterations result in saving of computational cost.

IV. EXPERIMENTAL RESULTS FOR BENCHAMARK FUNCTIONS

The well-known ten benchmark functions [5] in Table I are used to validate performance of the proposed technique (SASPSO). The authors are considered both unimodal and multimodal functions to test efficiency of the SASPSO. For this, the functions f_1 are taken for unimodal verification and f_2 to f_3 meant for multimodal. The global optima, search range and initialization range for each benchmark function is presented in Table II [5]. The stopping criteria of the proposed method is same as compared method.

The SASPSO has been tested on Rosenbrock, Rastrigin and Griewank benchmark functions by changing dimensions(d), generations(T) ,population size (P) with 30 trials as shown in Table III, IV and V. The results are compared with SSM-PSO[5] and LDW-PSO[5]. The low values of fitness obtained by SASPSO validate that, the proposed technique outperforms well than SSM-PSO [5] and LDW-PSO [5] for Rosenbrock, Rastrigin and Griewank benchmark functions. As seen from results the SASPSO improves the quality of optima as the number generation progresses. The presented method is suitable for optimization of unimodal and multimodal functions.

TABLE I. BENCHMARK FUNCTIONS FOR SIMULATIONS

Function Name	Mathematical Description
Rosenbrock f_1	$\sum_{i=1}^{d}(100(x_{i+1}-x_i^2)^2+(x_i-1)^2$
Rastrigin f_2	$\sum_{i=1}^{d}(x_i^2-10COS\ 2\pi x_i+10)$
Griewank f_3	$\frac{1}{4000}\sum_{i=1}^{d}x_i^2-\prod_{i=1}^{d}COS\frac{x_i}{\sqrt{i}}+1$

TABLE II. SEARCH RANGE AND INITILIZATIO RANGE FOR BENCHMARK FUNCTIONS

Function	Global Optima	Search Range	Initialization Range
Rosenbrock	0	$[-100, 100]^d$	$[15, 30]^d$
Rastrigin	0	$[-10, 10]^d$	$[2.56, 5.12]^d$
Griewank	0	$[-600, 600]^d$	$[300, 600]^d$

TABLE III. MEAN FITNESS VALUES FOR THE ROSENBROCK FUNCTIONS

Population	Dimensions	Iterations	SSM-PSO[5]	LDW-PSO[5]	SASPSO
20	10	1000	80.6875	95.1724	**36.5321**
	20	1500	112.9138	205.3715	**55.4256**
	30	2000	247.4600	307.4165	**69.5478**
40	10	1000	30.0598	68.1148	**32.5247**
	20	1500	74.8393	175.9682	**43.4512**
	30	2000	133.9569	296.6071	**63.4421**
80	10	1000	11.4541	37.1952	**28.3356**
	20	1500	36.0802	85.1608	**45.4125**
	30	2000	56.5048	200.4575	**55.2145**

TABLE IV. MAEN FITNESS VALUES FOR RASTRIGIN FUNCTIONS

Population	Dimensions	Iterations	SSM-PSO[5]	LDW-PSO [5]	SASPSO
20	10	1000	5.2552	5.6157	**0.4314**
	20	1500	21.3847	22.7886	**0.9965**
	30	2000	42.8655	47.1194	**6.4215**
40	10	1000	4.5459	3.6521	**0.7548**
	20	1500	18.6144	17.1412	**1.1245**
	30	2000	35.6870	38.6152	**4.4785**
80	10	1000	3.5159	2.4185	**0.4789**
	20	1500	12.2380	13.4634	**0.7854**
	30	2000	32.2035	30.3164	**3.4785**

TABLE V. MAEN FITNESS VALUES FOR GRIEWANK FNCTIONS

Population	Dimensions	Iterations	SSM-PSO[5]	LDW-PSO [5]	SASPSO
20	10	1000	0.1178	0.0919	**0.00875**
	20	1500	0.0267	0.0302	**0.00657**
	30	2000	0.0416	0.0179	**0.00314**
40	10	1000	0.0923	0.0861	**0.00658**
	20	1500	0.0302	0.0292	**0.00347**
	30	2000	0.0168	0.0127	**0.00247**
80	10	1000	0.0753	0.0772	**0.00095**
	20	1500	0.0272	0.0280	**0.00078**
	30	2000	0.0179	0.0301	**0.00035**

V. CONCLUSION

The author introduced Self Accelerated Smart Particle Swarm Optimization (SASPSO) to optimize multimodal optimization problems with fast convergence. In SASPSO, the positions of particle are updated by pbest (Personal or local best) and gbest (Global best). The main advantages of SASPSO are that it doesn't require velocity equation. In addition, it does not require any additional parameter like acceleration coefficients and inertia weight as in case other PSO algorithms. Momentum factor is introduced in SASPSO which can prevent particles from out of defined region without checking the validity of positions at every iterations result in saving of computational cost. The results of benchmark functions obtained by SASPSO validate that the proposed technique is more efficient for improving the quality of global optima of multimodal functions with less computational requirement.

REFERENCES

[1] Boeringer D.W. and Werner D.H., "Particle swarm optimization versus genetic algorithms for phased array synthesis", *IEEE Transactions Antennas Propagation*, vol. 52, no.3, pp.771–779, 2004.

[2] Eberhart R.C. and Shi Y., "Comparison between genetic algorithm and particle swarm optimization", In *proc. IEEE Int. Conf. Computt*., Anchorage, AK, pp.611-616, 1998.

[3] Kennedy J. and Eberhart R.C., "Particle Swarm Optimization" Proc York: *IEEE International conference on neural networks*, Springer-Verlag1995, Piscataway; vol.4, pp.1942-1948. 1985.

[4] Shi Y.H. And Eberhart R.C., "Parameter Selection in Particle Swarm Optimization" *Annual Conference on Evolutionary Computation*, pp.101-106, 1999.

[5] Liu Yu, Zheng Qin, and Xingshi He., "Supervisor-Student Model in Particle Swarm Optimization", *IEEE Congress on Evolutionary Computation 2004 (CEC 2004)*, vol.1, pp.542-547. 2004.

[6] Shi Y. and Eberhart R.C., "A modified particle swarm optimizer", Proceedings of the *IEEE Congress on Evolutionary Computation (CEC 1998)*, Piscataway, NI. pp.69-73. 1998.

[7] Pasupuleti Srinivas and Roberto Bhattiti, "The Gregarious Particle Swarm Optimizer (G-PSO)", *GECCO 2006*, Seattle, Washington, USA. 2006.

[8] Arumugam M. Senthil, Rao M. V. C., and Chandramohan Aarthi, "A new and improved version of particle swarm optimization algorithm with global-local best parameters" *Journal of Knowledge and Information System (KAIS), Springer*. vol.16, no.3, pp.324-350, 2008.

[9] Yang Ge and Rubo Zhang, "An Emotional Particle Swarm Optimization Algorithm", *Advances in Natural Computation, Lecture notes in Computer Science*, Springer-Verlag, Berlin, Germany, vol.3612, pp.553-561, 2005.