I.J. Intelligent Systems and Applications, **2015, 01, 62-72**
Published Online December 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2015.01.06

Adaptive Random Link PSO with Link Change Variations and Confinement Handling

Mrs. **Snehal Mohan Kamalapur**
Research Scholar, Dr. D Y Patil Institute of Engineering and Technology, Pune, University of Pune, Maharashtra, India
E-mail:snehal_kamalapur@yahoo.com

Dr. **Varsha Hemant Patil**
Research Guide, University of Pune, Maharashtra, India
E-mail: varsha.patil@gmail.com

Abstract— Particle Swarm Optimization is swarm based optimization technique. Swarm consists of particles and the particles fly through the problem space in Particle Swarm Optimization (PSO). Confinement methods and parameters such as Inertia Weight, Neighborhood of the particle have major impact on PSO performance. The paper presents variations of the PSO with adaptive random link neighborhood. The work carried out considers linearly decreasing inertia weight and different confinement methods. The performance of adaptive random link PSO by geometrical updation of velocity with confinement methods is tested here.

Index Terms—Adaptive Random Link, Confinement, Inertia Weight, Neighborhood, SPSO

I. INTRODUCTION

Particle Swarm Optimization (PSO) is an optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995 [1][2][3]. The swarm is analogous to social interaction. The particle in swarm is a learner as well as a guide. Every particle interacts and changes its position dynamically. The particles fly through problem space. A particle can be repositioned or its velocity can be modified to prevent it from leaving the search space. The repositioning of particle is done by using position confinement approach. Velocity of the particle is restricted in search space by using different velocity confinement methods.

Computational behavior of PSO is affected by parameter modification. Parameters include swarm size, acceleration coefficient and inertia weight or constriction coefficient. The parameters of the PSO are revised by parameter modification. Parameter setting is important as the performance of PSO is sensitive to the parameter settings.

Section II explains particle swarm optimization while section III and IV describe highlights on confinement methods and adaptation. Parameter control and Benchmark functions are briefly described in section V and VI respectively. Section VII focuses on experimental set up. Section VIII experimentally validates the correlation. The paper closes with conclusions in section IX.

II. PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization is swarm or population based optimization technique. The first version of the PSO was published by Kennedy and Eberhart in 1995 [1][2]and it has rapidly progressed in recent years since then. The particles move iteratively within the search space. Two main versions of PSO namely Global Best version and Local best version were introduced in early phase of PSO research. Global best is the best position achieved by any of the particle in the swarm. Global best particle is influential and influences to update position of all particles. Local best is the best position achieved in the neighborhood of a particle.

Swarm of N particles is represented as-

$S = \{x_1, x_2, ..., x_N\}$

For a D-dimensional search space, the position of the i[th] particle is represented as:

$x_i = (x_{i1}, x_{i2}, ..., x_{iD})^T$ where i = 1,2,...,N

Let t denote iteration counter. So $x_i(t)$ and $v_i(t)$ denote current position and velocity of i[th] particle respectively.

The position of each particle is updated using a proper position shift. This is called velocity and is denoted as

$v_i = (v_{i1}, v_{i2}, ..., v_{iD})^T$ where i = 1,2,...,N

A. Algorithm of Global Best version of PSO

- Initialize the population of particles randomly Do
- Calculate fitness values of each particle as per objective function
- Update particle's best position if the current fitness value is better than earlier best position
- Determine the best fitness value in the swarm
- Update velocity of each particle using (1)

$$v_i(t+1) = v_i(t) + c_1 * r_1 * (p_i - x_i) + c_2 * r_2 * (p_g - x_i) \quad (1)$$

- Update position of particle using equation (2)

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

While maximum number of iterations or criterion is not attained

B. Algorithm of Local best version of PSO

- Initialize the population of particles randomly Do
- Calculate fitness values of each particle as per objective function
- Update particle's best position if the current fitness value is better than earlier best position
- Determine the best fitness value in the swarm
- Update velocity of each particle using (3)

$$v_i(t+1) = v_i(t) + c_1 * r_1 * (p_i - x_{id}) + c_2 * r_2 * (p_l - x_i) \quad (3)$$

- Update position of particle using equation (2)

While maximum number of iterations or criterion is not attained

C. Standard PSO 2011

Three Standard PSO algorithms have been defined in the literature of PSO [4], [5], [6], [7], [8], [9]. Until 2007, the velocity update was carried out dimension by dimension method. It is known that dimension by dimension method is biased [10]. When the optimum point lies on an axis, on a diagonal or on the centre of the system of coordinates then it is easy to find it for PSO. Spear analyzed this in 2010 [11]. Clerc proposed SPSO2011 as a solution to this bias [5]. Standard PSO (SPSO-2011) [8], [9] exploits the idea of rotational invariance. SPSO-2011 modifies velocity in geometrical way.

Let G_i (t) be the centre of gravity of the current position, a point a bit beyond the best previous position and a point a bit beyond the best previous position in the neighborhood.

Center of gravity is calculated using (4)-

$$G_i(t) = x_i + c * \frac{p_i + l_i - 2*x_i}{3} \quad (4)$$

Let x_i' be random point defined using uniform distribution in the hypersphere of radius G_i and center (G_i-x_i) that is $H_i(G_i, \|G_i - x_i\|)$

Velocity of the particle is then calculates using (5)

$$v_i(t+1) = \omega * v_i(t) + x'_i(t) - x_i(t) \quad (5)$$

Position of the particle is calculates using (2).

III. Parameters of PSO

- *Swarm Size :*

The Swarm size is a size of the population. The swarm consists of particles. Swarm size varies from 20 to 60. Ideally swarm size is considered as 40. The swarm size can also be computed using (6) –

$$s = 10 + [2 * sqrt(D)] \quad (6)$$

- *Velocity Threshold*

The first issue observed and addressed by researchers is swarm explosion. Solution put forth by researcher was velocity clamping. Velocity clamping prevents particles uncontrolled increase of magnitude from current positions. Whenever velocity goes beyond threshold bound it is directly set to closest bound or threshold. Details of velocity confinement are discussed in section IV.

- *Inertia Weight*

Inertia weight was not used in the first version of PSO developed by Eberhart. As swarm was not able to converge towards promising position with velocity clamping, a new parameter inertia weight ω was introduced [12] in the equation (1) as shown in (7)

$$v_{id}(t+1) = \omega * v_{id}(t) + c_1 * r_1 * (p_{id} - x_{id}) + c_2 * r_2 * (p_{gd} - x_{id}) \quad (7)$$

The inertia weight defines the impact of previous velocity of each particle to the current one and controls the scope of search. Poli et al. [13] interpreted it as the fluidity of the medium in which particles move.

Many variations of PSO were proposed based on inertia weight. The Linearly Decreasing[12][14], Linearly increasing [15] , Nonlinear [16], Sigmoid decreasing [17] , Adaptive [18], Random [19], Chaotic [20], Oscillating [21], Logarithmic decreasing [22] , Exponent Decreasing inertia weight strategies [23], [24], [25] and Fuzzy Adaptive Inertia Weight [26] were used in the literature. The Linearly Decreasing strategy improves the efficiency and performance of PSO.

- *Acceleration Coefficient*

Acceleration coefficients c_1 and c_2 indicate cognitive and social influence values respectively.

- *Neighborhood Topology*

Neighborhood topology is a scheme to determine the neighbors of particles in a swarm. Information exchange among the particles is related to exploration capability of the swarm. Each particle may have set of other particles as neighbors.

The actual distance of the particles can be calculated to form neighborhood. But this requires (N (N+1))^2 computations at each iteration, if N is size of the swarm. Simple alternate solution to this is to use indices of particles to decide neighborhood. The cardinality of neighborhood called neighborhood size depends on type of neighborhood topology.

In PSO, each particle has communication neighborhood so several studies were performed in order to determine effect of the neighborhood topology on the convergence.

The performance of PSO can be improved by selecting proper neighborhood topology.

The neighborhood topology can be classified as [27], [28]-

- gbest Topology
- Ring Topology
- Adaptive Random
- Mesh Topology
- Tree or hierarchical Topology
- Toroidal Topology
- Dynamic Topology

- lbest Ring Lattice with dynamic increment neighborhood
- Fitness Distance Ratio
- Random Edge Migration
- Dynamic Hierarchy
- TRIBES

Some topologies are good for global optimization while some topologies are good for local optimization.

IV. Confinements of PSO

A particle tends to leave the search space in Particle Swarm Optimization. A particle can be repositioned or velocity can be modified to prevent it from leaving the search space [29]. These two types of confinement are discussed here.

A. Prevention

Whenever velocity goes beyond threshold bound one of the following confinement is used for Velocity of Particle in the literature-

1. Absorb or Zero

$$if\ v(t+1) > v_{max} \quad or\ v(t+1) < v_{max}$$
$$then\ v(t+1) = 0 \tag{8}$$

2. Deterministic Back

$$if\ v(t+1) > v_{max} \quad or\ v(t+1) < v_{max}$$
$$v(t+1) = -\lambda * v(t+1) \tag{9}$$

Where $\boldsymbol{\lambda}$ can be a predefined value.

3. Random back

If velocity goes beyond threshold bound the velocity is reversed back by multiplying with random value drawn from uniform distribution [0 1]

4. Adjust

If velocity goes beyond threshold bound it is adjusted using position of the particle-

$$\mathrm{v(t+1) = x(t+1) - x(t)} \tag{10}$$

5. Hyperbolic

If velocity goes beyond threshold bound it is normalized as

$$v(t+1) = \frac{v(t+1)}{1 + |\frac{v(t+1)}{x_{max} - x(t)}|} \quad if\ v(t+1) > 0 \tag{11}$$

$$v(t+1) = \frac{v(t+1)}{1 + |\frac{v(t+1)}{x(t) - x_{min}}|} \quad if\ v(t+1) < 0 \tag{12}$$

B. Repositioning Particle

Whenever position of the particle goes beyond threshold bound one of the following confinement is used -

1. Nearest

When the position of the particle moves away from the boundary the particle is adjusted using (13)(14)

$$if\ x(t+1) > x_{max} \quad then\ x(t+1) = x_{max} \tag{13}$$

$$if\ x(t+1) < x_{min} \quad then\ x(t+1) = x_{min} \tag{14}$$

2. Do not Change

When the position of the particle moves away from the boundary the particle is adjusted using (15)

$$if\ x(t+1) > x_{max}\ or\ \ x(t+1) <\ x_{min}\ \ then$$
$$x(t+1) = \ x(t+1) - \ v(t+1) \tag{15}$$

3. Reflex

When the position of the particle moves away from the boundary the particle is adjusted using (16)(17)

$$if\ x(t+1) > x_{max} \quad then\ x(t+1) = x_{max} - (x_{max} - x(t+1)) \tag{16}$$

$$if\ x(t+1) < x_{min} \quad then\ x(t+1) = x_{min} + (x_{min} - x(t+1)) \tag{17}$$

V. Parameter Control

The parameter control can be used depending on how and what is changed. Parameters can be controlled using deterministic rule or by adaptation. The deterministic rule is used to control the parameter value for all iterations. This gives better performance for some cases but not in all cases. Hence the parameter control can be done using current state of the search.

Adaptation used here is based on two parameters-

1. Inertia Weight

The searching varies from exploratory phase, towards the refinement of local search at the end. Hence deterministic approach is used for inertia weight. A linearly decreasing inertia weight strategy decreases the value of inertia weight with generation number. An inertia weight value is set to 0.9 initially during explorative stage and linearly decremented to 0.4.

2. Population structure

Adaptive Random topology has been defined in [30] informs K randomly chosen particles in swarm. . Generally K is set to 3. Link modification is adaptive and depends on the fitness values at swarm level. The information links that is K particles are selected at the beginning, and after unsuccessful iteration. Variation of this adaptive random link is tested here. Instead of changing informants after unsuccessful iteration, informants are changed after some threshold number of unsuccessful iterations.

VI. Benchmark Functions

Both unimodal as well as multimodal benchmark functions [9] are chosen for experimentation and are listed below-

- *The Sphere Function*

The Sphere function is Continuous, Differentiable, Separable, Scalable and Multimodal, defined as follows-

$$f_1 = \sum_{i=1}^{D} x_i^2 \tag{18}$$

- *Rastrigin Function*

Rastrigin is highly multimodal and the location of the minima are regularly distributed. It is defined as -

$$f_2 = \sum_{i=1}^{D}[x_i^2 - 10\cos(2\pi x_i) + 10] \tag{19}$$

- *Step Function*

Step function is Discontinuous, Non-Differentiable, Separable, Scalable and Unimodal, , defined as follows -

$$f_3 = \sum_{i=1}^{D}(\lfloor x_i + 0.5\rfloor)^2 \tag{20}$$

- *Rosenbrock's Function*

Rosenbrock's function is Continuous, Differentiable, Non-Separable, Scalable and Unimodal, defined as follows:

$$f_4 = \sum_{i=1}^{D}[100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2] \tag{21}$$

Where D is the dimension and

$x = (x_1, x_2, \cdots; x_D)$ is D-dimensional row vector.

VII. Experimental Setup

The local best PSO is used for experimental tests and parameters are set as discussed below -

- *Initial population*

The positions for a particle in the swarm are initialized using uniform distribution along each dimension of the problem space.

- *Swarm Size*

Swarm size is considered as 40.

- *Inertia Weight*

An inertia weight value starting from 0.9 linearly decreasing to 0.4 improves the performance [12], [14].

- *Velocity*

Initially velocity is set to zero

- *Acceleration coefficient*

c1 and c2 are set to 1.4.

- *Neighborhood Topologies*

In PSO, each particle has communication neighborhood. Following neighborhood topologies are used -

  - Adaptive Random Link
  - Adaptive Random Link with link change after threshold number of unsuccessful iterations.

- *Bound Handling*

As discussed by Clerc Hyperbolic and random back give better results. Hence the work considers these two prevention methods and all repositioning methods are verified with these two prevention methods for adaptive random link.

Velocity Bound Handling

1: Random back

2: Hyperbolic

Position Bound Handling

1: Nearest

2: Do not change position

3: Reflex

The variants used for experiments are –

1. Adaptive random link(RL) with link change after unsuccessful iterations

Adaptive random link(RL) with links change after 10, 20, 30,…., 100 unsuccessful iterations lc10, lc20, …, lc100 represents Link Change after 10 unsuccessful iterations, 20 unsuccessful iterations,…., 100 unsuccessful iterations respectively.

Table 1. The variants used for experiments

	Neighbourhood Topologies	Velocity confinement	Position confinement
RL11	Adaptive random link	Random back	Nearest
RL12			Do not change
RL13			Reflex
RL21		Hyperbolic	Nearest
RL22			Do not change
RL23			Reflex

The procedure is iterative. Stopping criteria is maximum number of iterations (5000) or acceptable error between optimal solution known and calculated as 10E-20.

VIII. Results and Discussion

The experiments are carried out on both unimodal and multimodal functions. Ten runs are taken for each. Average number of iterations required to obtain acceptable solution are summarized in table 2, 3, 4, 5, 6 and 7. Bold faced values represent minimum number of iterations required for link change after threshold number of unsuccessful iterations (Row minimum).

Table 2. Average number of iterations for random link Threshold (threshold ranging from 10 to 100) for two dimensional Sphere function

Link change threshold	RL11	RL12	RL13	RL21	RL22	RL23
1	573	**536**	547	575	537	571
10	564	527	535	571	549	**524**
20	599	552	557	547	**534**	541
30	**557**	580	603	592	563	593
40	553	560	**549**	578	576	559
50	611	574	581	**552**	624	564
60	**529**	596	562	563	560	573
70	566	626	**534**	579	555	535
80	**559**	594	592	608	565	563
90	607	582	599	611	585	**580**
100	594	593	574	574	616	**553**

Table 3. Average Number Of Iterations For Random Link Threshold (Threshold Ranging From 10 To 100) For Two Dimensional Rastrigin Function

Link change threshold	RL11	RL12	RL13	RL21	RL22	RL23
1	**1846**	1957	1959	2021	2087	2046
10	1970	1815	1800	1761	1867	**1746**
20	1944	1946	1988	1760	1804	**1669**
30	2013	1627	1888	**1542**	1852	1827
40	**979**	1418	1126	1133	1092	1209
50	1001	1301	**921**	1137	994	967
60	1055	1081	**857**	1103	943	954
70	1019	1020	**993**	1021	1007	1100
80	1056	1034	1081	**815**	877	985
90	938	1059	1069	969	1027	**935**
100	1066	962	**869**	1134	1045	1049

Table 4. Average number of iterations for random link Threshold (threshold ranging from 10 to 100) for two dimensional Step function

Link change threshold	RL11	RL12	RL13	RL21	RL22	RL23
1	**37**	38	41	48	44	46
10	47	46	44	**37**	49	40
20	54	**38**	44	48	46	46
30	51	47	**35**	47	46	57
40	47	37	45	53	47	**36**
50	43	50	42	42	**38**	45
60	43	40	45	44	49	**37**
70	56	52	49	42	**41**	47
80	49	47	45	48	45	**38**
90	45	47	53	56	45	**43**
100	48	43	**31**	**43**	43	37

Table 5. Average number of iterations for random link Threshold (threshold ranging from 10 to 100) for two dimensional Rosenbrock function

Link change threshold	RL11	RL12	RL13	RL21	RL22	RL23
1	3471	3427	3418	**3239**	3311	3480
10	3540	3589	3447	**3371**	3457	3372
20	3425	**3327**	3416	3425	3477	3484
30	3181	3401	**3015**	3159	3097	3191
40	2509	**2485**	2843	2859	2870	3040
50	2504	2077	2179	**2029**	2372	2153
60	**1944**	2207	2075	1948	2077	2089
70	2087	**2081**	2136	2138	2166	2131
80	2206	2065	2043	2087	2040	**2037**
90	2127	2191	**2031**	2224	2112	2151
100	2196	**2069**	2182	2271	2213	2182

The variations in the average number of iterations required for adaptive random link variants with threshold ranging from 1 to 100 is shown in fig. 1.

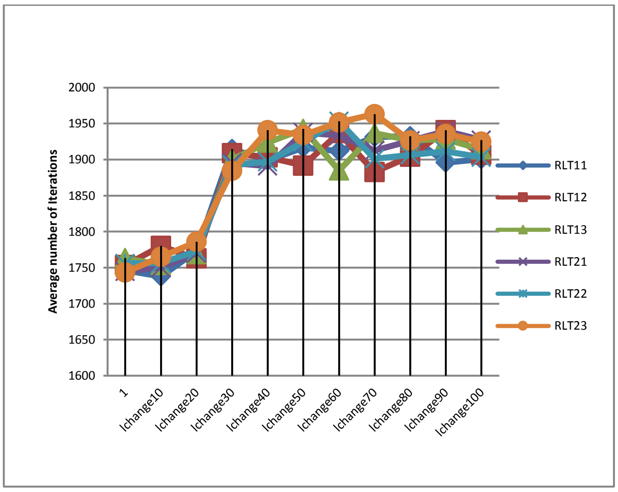

Fig.1. Average number of iterations for random link Threshold (threshold ranging from 10 to 100) for ten dimensional Sphere function

Table 6. Average number of iterations for random link Threshold (threshold ranging from 10 to 100) for ten dimensional Sphere function

Link change threshold	RL11	RL12	RL13	RL21	RL22	RL23
lc1	1746	1753	1763	1745	1756	**1744**
lc10	**1738**	1780	1754	1750	1756	1765
lc20	1773	**1763**	1769	1770	1773	1786
lc30	1915	1909	1900	1896	1892	**1884**
lc40	1899	1903	1924	**1891**	1897	1941
lc50	1917	**1892**	1942	1938	1925	1934
lc60	1912	1937	**1885**	1933	1953	1952
lc70	1931	**1883**	1937	1913	1901	1963
lc80	1932	**1904**	1927	1926	1907	1927
lc90	**1896**	1941	1928	1940	1911	1936
lc100	**1901**	1907	1914	1927	1903	1925

Table 7. Average number of iterations for random link Threshold (threshold ranging from 10 to 100) for ten dimensional Step function

Link Change threshold	RL11	RL12	RL13	RL21	RL22	RL23
lc1	772	768	**721**	798	778	786
lc10	792	758	789	828	**747**	804
lc20	**729**	783	794	759	762	779
lc30	864	**794**	849	822	841	808
lc40	**823**	865	847	859	837	850
lc50	944	900	946	937	891	**873**
lc60	**851**	932	864	877	862	926
lc70	896	869	**867**	899	929	941
lc80	907	929	909	939	943	**890**
lc90	957	943	988	908	901	**873**
lc100	915	939	916	917	955	**902**

Table 8 and 9 focuses on minimum value obtained for benchmark function. Standard deviation shows variation for type of functions and link change threshold. Tables 10, 11, 12 and 13 Statistical evaluation.

Table 8. Min Value Obtained (Threshold Ranging From 1 To 100) For Ten Dimensional Sphere Function

Link change threshold	RLT11	RLT12	RLT13	RLT21	RLT22	RLT23
1	8.84E-21	7.9E-21	6E-21	8.41E-21	2.1E-21	6.6E-21
10	8.24E-21	6.1E-21	6.3E-21	4.96E-21	5.9E-21	4.6E-21
20	5.11E-21	6.5E-21	6.3E-21	7.06E-21	6.6E-21	7.1E-21
30	6.54E-21	4.5E-21	5.9E-21	6.85E-21	4.7E-21	7.6E-21
40	7.6E-21	8.3E-21	7.2E-21	5.47E-21	5.5E-21	7.1E-21
50	5.11E-21	7.1E-21	7.9E-21	5.98E-21	5.6E-21	7.1E-21
60	5.95E-21	2.7E-21	4.9E-21	5.54E-21	4.4E-21	5.7E-21
70	5.21E-21	8.1E-21	7.3E-21	4.9E-21	7.3E-21	6.5E-21
80	4.36E-21	5.7E-21	5.3E-21	6.21E-21	5.8E-21	8.5E-21
90	3.85E-21	4.7E-21	7.1E-21	4.19E-21	3.1E-21	7.5E-21
100	7.59E-21	7.5E-21	4.8E-21	7.67E-21	6.2E-21	7.8E-21

Table 9. Min Value Obtained (Threshold Ranging From 1 To 100) For Ten Dimensional Step Function

Link change threshold	RLT 11	RLT 12	RLT 13	RLT 21	RLT 22	RLT 23
1	0	0	0	0	0	0
10	0	0	0	0	0	0
20	0	0	0	0	0	0
30	0	0	0	0	0	0
40	0	0	0	0	0	0
50	0	0	0	0	0	0
60	0	0	0	0	0	0
70	0	0	0	0	0	0
80	0	0	0	0	0	0
90	0	0	0	0	0	0
100	0	0	0	0	0	0

Table 10. Statistical Evaluation for Two Dimensional Sphere Function Based On Average Number for Iterations

Link change threshold	Min.	Median	Mean	Max.	Std. Deviation
1	536	559	557	575	19
10	524	542	545	571	20
20	534	550	555	599	23
30	557	586	581	603	18
40	549	560	563	578	12
50	552	578	584	624	28
60	529	563	564	596	22
70	534	561	566	626	34
80	559	579	580	608	20
90	580	592	594	611	13
100	553	584	584	616	22

Interpretation of correlation coefficient –

Correlation analysis is used to study the interdependence of two variables. If large (small) values are associated with large (small) values of other variable then there exists strong positive correlation. If large (small) values are associated with small (large) values of other variable then there exists strong negative correlation.

1. If correlation coefficient is 1 then perfect positive linear relationship exists between two variables.
2. If correlation coefficient is -1 then perfect negative linear relationship exists between two variables
3. If correlation coefficient is greater than or equal to 0.7 and less than or equal to 1 then strong positive linear relationship exists between two variables
4. If correlation coefficient is greater than or equal to -1 and less than or equal to -0.7 then strong negative linear relationship exists between two variables
5. If correlation coefficient is 0 then no linear relationship between two variables exists between two variables

Table 11. Statistical evaluation for two dimensional Rastrgin function based on average number for iterations

Link change threshold	Min.	Median	Mean	Max.	std dev
1	536	559	557	575	85
10	524	542	545	571	82
20	534	550	555	599	127
30	557	586	581	603	175
40	549	560	563	578	147
50	552	578	584	624	141
60	529	563	564	596	96
70	534	561	566	626	38
80	559	579	580	608	106
90	580	592	594	611	60
100	553	584	584	616	92

Table 12. Statistical Evaluation for Two Dimensional Step Function Based On Average Number for Iterations

Link change threshold	Min.	Median	Mean	Max.	Std. Deviation
1	37	43	42	48	4
10	37	45	44	49	5
20	38	46	46	54	5
30	35	47	47	57	7
40	36	46	44	53	7
50	38	43	43	50	4
60	37	44	43	49	4
70	41	48	48	56	6
80	38	46	45	49	4
90	43	46	48	56	5
100	31	43	41	48	6

Table 13. Statistical evaluation for two dimensional Rosenbrock function based on average number for iterations

Link change threshold	Min.	Median	Mean	Max.	std dev.
1	3239	3422	3391	3480	96
10	3371	3452	3463	3589	88
20	3327	3425	3426	3484	56
30	3015	3170	3174	3401	129
40	2485	2851	2768	3040	222
50	2029	2166	2219	2504	183
60	1944	2076	2057	2207	99
70	2081	2134	2123	2166	33
80	2037	2054	2080	2206	65
90	2031	2139	2139	2224	67
100	2069	2189	2186	2271	66

Table 14. Correlation For Two Dimensional Sphere Function With Link Change From 1 To 100.

Link change threshold	RL11	RL12	RL13	RL21	RL22	RL23
1	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
10	**0.99**	**0.86**	**0.98**	**0.89**	**0.96**	**0.88**
20	-0.10	**0.76**	0.61	-0.03	**0.89**	0.41
30	0.59	0.27	0.02	0.31	-0.44	-0.24
40	0.28	**0.89**	-0.13	0.71	0.44	0.32
50	0.25	0.16	0.12	0.68	0.38	**0.82**
60	0.61	-0.22	-0.18	-0.37	0.52	0.51
70	0.15	0.25	0.06	0.60	-0.01	-0.23
80	0.26	0.26	-0.58	NA	NA	0.05
90	0.13	0.01	-0.15	NA	0.55	0.63
100	NA	0.22	NA	-0.05	NA	0.15

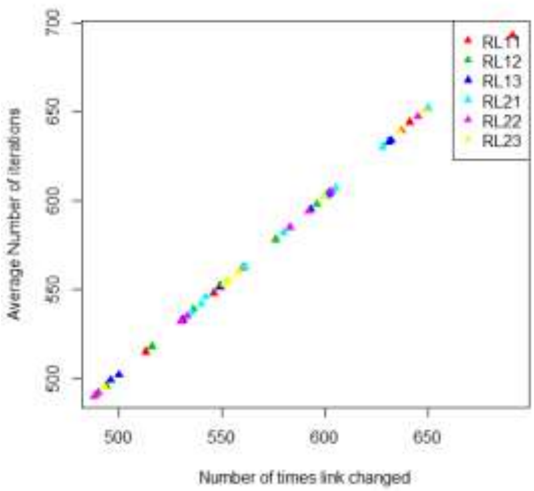

Fig. 2. Average number of iterations vs Number of times link changed for sphere Function D2 link change after every unsuccessful iteration

Figures 2 to 12 show plots of average number of iterations against number of times link changed for benchmark functions depending on iterations.

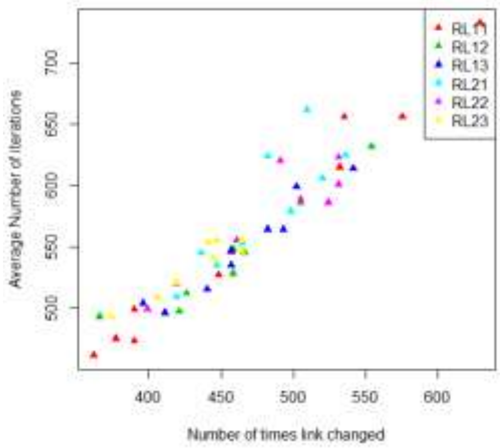

Fig.3. Average number of iterations vs Number of times link changed for sphere Function D2 link change after 10 unsuccessful iteration

Table 15. Correlation For Two Dimensional Rastrigin Function With Link Change From 1 To 100.

Link change threshold	RL11	RL12	RL13	RL21	RL22	RL23
1	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
10	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
20	**1.00**	**0.98**	**0.99**	**1.00**	**1.00**	**0.99**
30	**0.99**	**1.00**	**1.00**	**0.99**	**0.99**	**0.99**
40	**0.83**	**0.99**	**0.96**	**0.86**	**0.90**	**0.92**
50	**0.98**	**0.92**	**0.91**	**0.89**	**0.86**	**0.92**
60	**0.84**	**0.93**	**0.74**	**0.94**	**0.83**	**0.84**
70	**0.85**	**0.86**	**0.93**	**0.88**	**0.62**	**0.74**
80	**0.81**	**0.85**	0.08	**0.82**	0.63	**0.95**
90	**0.89**	0.63	0.60	**0.92**	**0.91**	**0.80**
100	**0.82**	**0.76**	**0.90**	**0.77**	**0.88**	0.34

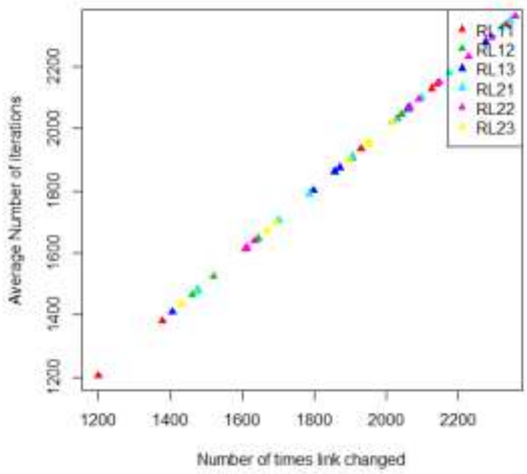

Fig. 4. Average number of iterations vs Number of times link changed for Rastrigin Function D2 link change after every unsuccessful iteration

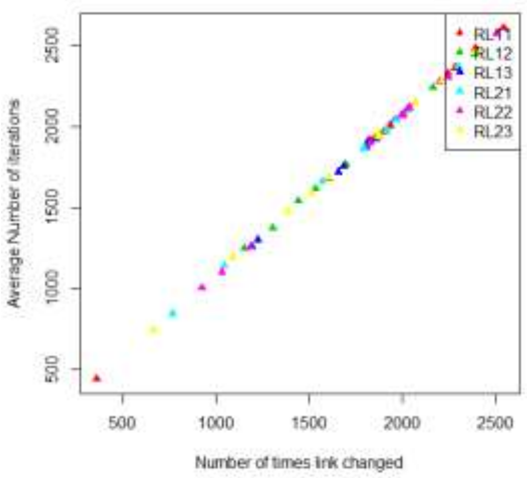

Fig. 5. Average number of iterations vs Number of times link changed for two dimensional Rastrigin Function link change after 10 unsuccessful iteration

Table 16. Correlation for Two Dimensional Step Function

Link change threshold	RL11	RL12	RL13	RL21	RL22	RL23
1	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
10	**0.89**	**0.79**	**0.88**	**0.93**	**0.97**	**0.98**
20	0.66	0.48	**0.93**	**0.85**	**0.92**	**0.75**
30	0.65	**0.84**	NA	**0.95**	**0.71**	**0.73**
40	NA	NA	**0.79**	0.64	NA	**0.83**
50	NA	**0.71**	NA	NA	NA	NA
60	NA	NA	NA	NA	NA	NA
70	NA	NA	NA	NA	NA	NA
80	NA	NA	NA	NA	NA	NA
90	NA	NA	NA	NA	NA	NA
100	NA	NA	NA	NA	NA	NA

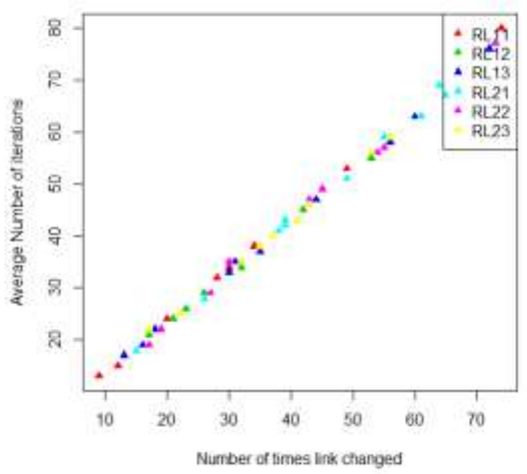

Fig. 6. Average number of iterations vs Number of times link changed for Step Function D2 link change after every unsuccessful iteration

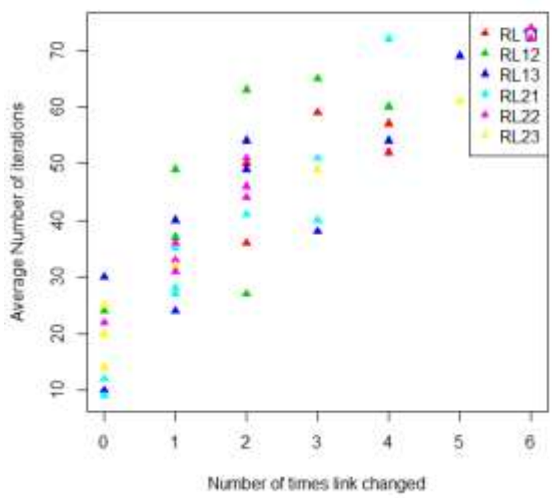

Fig.7. Average number of iterations vs Number of times link changed for Step Function D2 Link change =10

Table 17. Correlation for Two Dimensional Rosenbrock Function

Link change threshold	RL11	RL12	RL13	RL21	RL22	RL23
1	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
10	**0.91**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
20	**0.99**	**1.00**	**1.00**	**0.97**	**0.99**	**0.96**
30	**0.86**	**0.93**	**0.98**	**0.95**	**0.99**	**0.97**
40	**0.98**	**0.97**	**0.97**	**0.90**	**0.97**	**0.94**
50	**0.96**	**0.80**	**0.92**	**0.76**	**0.98**	**0.89**
60	0.64	**0.76**	0.64	0.64	0.13	**0.80**
70	0.60	0.57	**0.74**	**0.77**	**0.98**	0.48
80	**0.88**	**0.92**	0.68	0.31	0.55	0.40
90	0.62	0.59	0.59	**0.89**	**0.80**	**0.73**
100	0.45	0.57	0.65	0.30	0.60	**0.73**

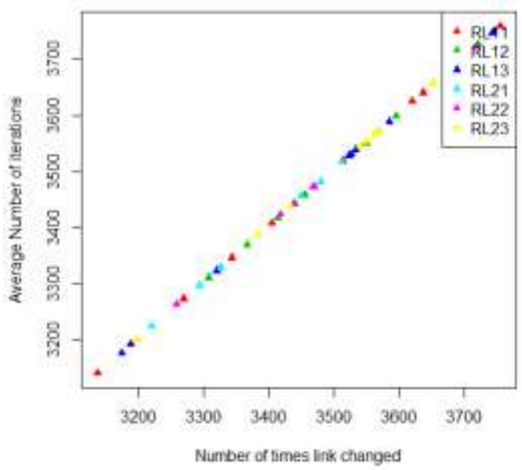

Fig. 8. Average number of iterations vs Number of times link changed for Rosenbrock Function D2 Link change =1

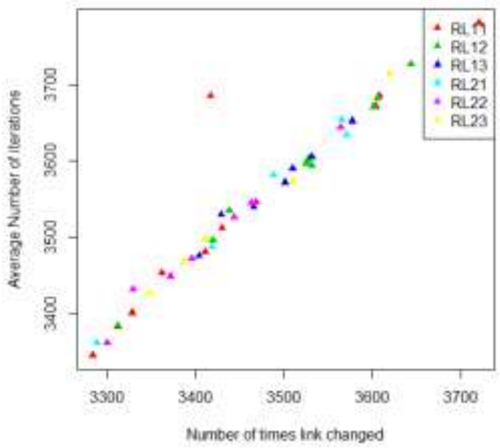

Fig.9. Average number of iterations vs Number of times link changed for Rosenbrock Function D2 with link change threshold=10

Table 18. Correlation For Ten Dimensional Sphere Function With Link Change From 20 To 100

Link Change threshold	RL11	RL12	RL13	RL21	RL22	RL23
1	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
10	**0.95**	**0.87**	**0.90**	**0.65**	**0.98**	**0.79**
20	0.47	0.29	0.41	0.32	0.30	0.47
30	0.09	-0.21	0.18	0.20	**0.78**	-0.31
40	0.07	0.37	-0.02	-0.45	0.09	-0.44
50	0.16	-0.55	0.06	-0.27	0.27	-0.47
60	0.22	-0.15	-0.28	-0.15	0.56	-0.55
70	**0.79**	0.48	0.17	-0.08	0.32	0.69
80	0.25	0.27	0.55	0.59	0.02	0.16
90	0.35	**0.81**	0.01	-0.49	-0.44	-0.06
100	0.51	**0.73**	-0.63	-0.32	0.49	NA

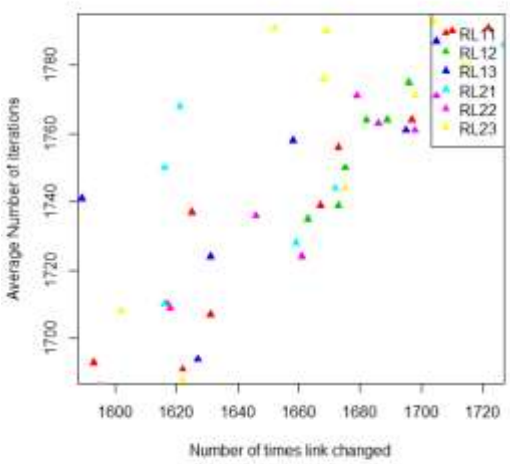

Fig. 10 Average number of iterations vs Number of times link changed for Sphere Function D10 with link change threshold=10

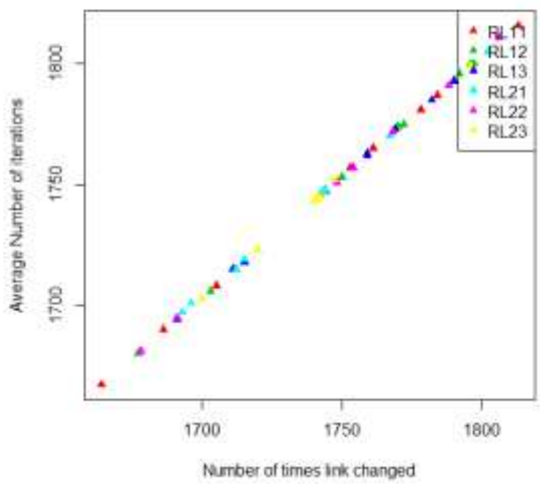

Fig. 11. Average number of iterations vs Number of times link changed for Sphere Function D10 with link change threshold=1

Table 19. Correlation For Ten Dimensional Step Function With Link Change From1 To 100

Link change threshold	RL11	RL12	RL13	RL21	RL22	RL23
1	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
10	**0.98**	**0.99**	**0.98**	**0.97**	**0.97**	**0.99**
20	**0.78**	**0.75**	**0.87**	**0.77**	**0.86**	**0.61**
30	**0.74**	**0.78**	0.51	0.38	**0.75**	0.55
40	**0.74**	**0.75**	0.23	**0.75**	0.64	**0.88**
50	**0.79**	**0.75**	0.39	0.60	0.54	0.67
60	**0.73**	**0.88**	0.40	**0.71**	**0.74**	0.49
70	0.36	**0.83**	0.60	0.10	0.64	**0.70**
80	**0.77**	**0.92**	0.38	0.49	0.46	0.45
90	0.59	0.61	0.42	**0.73**	-0.01	0.53
100	0.37	**0.89**	0.13	0.42	0.47	0.55

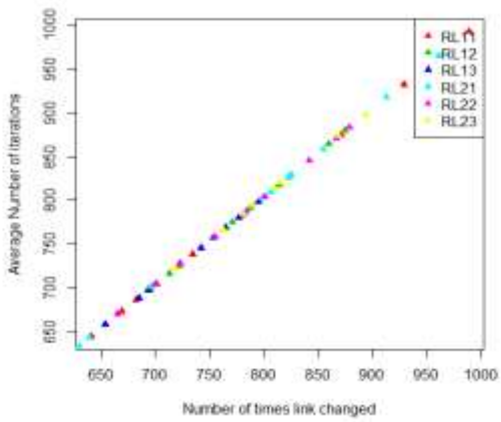

Fig. 12 Average number of iterations vs Number of times link changed for Step Function D10 with link change threshold=1

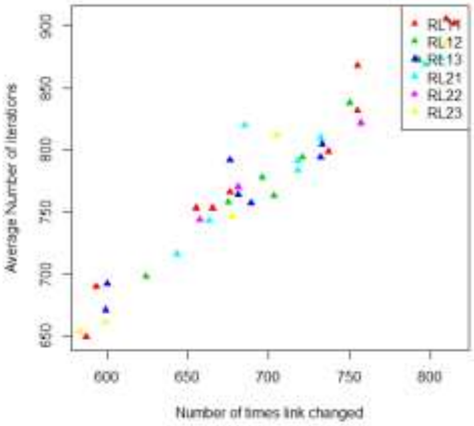

Fig.13. Average number of iterations vs Number of times link changed for Sphere Function D10 with link change threshold=10

Correlation is calculated using “Pearson” method for Link changes and all variants. Tables 14, 15, 16 17, 18 and 19 show correlation calculated using “Pearson” method. Bold faced values represent perfect or strong correlation.

NA: Not Applicable (As number of times link changed during progress is 0 for some threshold values, So if all values are same then the mean value is same and there is no deviation obviously).

Table 20. Correlation Observations

Link change threshold	Sphere			Rastrigin		
	perfect	strong	weak	perfect	strong	weak
1	6	0	0	6	0	0
10	0	6	0	6	0	0
20	0	2	4	3	3	0
30	0	0	6	2	4	0
40	0	2	4	0	6	0
50	0	1	5	0	6	0
60	0	0	6	0	6	0
70	0	0	6	0	5	1
80	0	0	4	0	4	2
90	0	0	5	0	4	2
100	0	0	3	0	5	1

Table 21. Correlation Observations

Link change threshold	Step			Rosenbrock		
	perfect	Strong	weak	perfect	strong	weak
1	6	0	0	6	0	0
10	0	6	0	5	1	0
20	0	4	2	2	4	0
30	0	4	1	0	6	0
40	0	2	1	0	6	0
50	0	1	0	0	6	0
60	0	0	0	0	2	4
70	0	0	0	0	3	3
80	0	0	0	0	2	4
90	0	0	0	0	3	3
100	0	0	0	0	1	5

If the link is updated after unsuccessful iteration then number of iterations required versus number of times link changed has perfect correlation over runs. Other findings are listed in table 20 and 21.

IX. CONCLUSION

Particle Swarm Optimization is swarm or population based optimization technique. To handle position and velocity confinement, different bound handling methods are used. Fixed pattern for average number of iterations required for type of bound handling methods and link change variations is not observed. But the standard deviation is more when link is changed after 30 to 50 unsuccessful iterations. Adaptive random neighborhood PSO with varying link structure after threshold number of unsuccessful iterations is tested. The experiments carried out using benchmark functions comprising of unimodal, multimodal, separable and non separable functions. The perfect positive correlation is observed in all variations for Adaptive random neighborhood PSO with varying link structure after unsuccessful iteration while the strong positive correlation is observed in all variations for Adaptive random neighborhood PSO with varying link structure after ten unsuccessful iterations. When the threshold number of unsuccessful iterations increases, correlation value moves towards weak.

ACKNOWLEDGMENT

The authors wish to thank the Principal and Research Coordinator of Padamashree Dr. D. Y. Patil Institute of Engineering and Technology, Pune.

REFERENCES

[1] J. Kennedy, R.C. Eberhart, “Particle swarm optimization,” Proceedings of the IEEE International Conference on Neural Networks, Piscataway, NJ, USA, 1995, Vol. 4, pp. 1942 – 1948

[2] R.C. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory", Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995, pp. 39 – 43

[3] J. Kennedy, R. C. Eberhart, and Y. H. Shi, Swarm Intelligence.San Mateo, CA: Morgan Kaufmann, 2001.

[4] Maurice Clerc, “A method to improve Standard PSO,” Tech. Rep. MC2009, pp. 03-13, 2009, http://clerc.maurice.free.fr/pso/Design efficient PSO.pdf

[5] Maurice Clerc, “Standard Particle Swarm Optimisation,” Particle Swarm Central, Tech. Rep., 2012, http://clerc.maurice.free.fr/pso/SPSO descriptions. pdf

[6] Maurice Clerc, “Beyond standard particle swarm optimisation,” International Journal of Swarm Intelligence Research, vol. 1, no. 4, pp. 46–61, 2010.

[7] Zambrano-Bigiarini, M. Clerc, Rojas, “Standard Particle Swarm Optimisation 2011 at CEC-2013: A baseline for future PSO improvements”. IEEE Congress on Evolutionary Computation (CEC) 2013 ,Digital Object Identifier: 10.1109/CEC.2013.6557848 , Page(s): 2337 – 2344

[8] PSC, "Particle Swarm Central," 2013, http://www.particleswarm.info/.
[9] Momin Jamil and Xin-She Yang, A literature survey of benchmark functions for global optimization problems, Int. Journal of Mathematical Modelling and Numerical Optimisation}, Vol. 4, No. 2, pp. 150--194 (2013)
[10] Christopher K. Monson and Kevin D. Seppi. Exposing Origin-Seeking Bias in PSO. In GECCO'05, pages 241_248, Washington, DC, USA, 2005.
[11] William M. Spears, Derek T. Green, and Diana F. Spears. Biases in particle swarm optimization. International Journal of Swarm Intelligence Research, 1(2):34_57, 2010
[12] Yuhui Shi , Eberhart, R, "A modified particle swarm optimizer ", Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence, 4-9 May 1998 PP 69 - 73
[13] R. Poli, J. Kennedy, and T. Blackwell. Particle swarm optimization. Swarm Intelligence, 1(1), 2007, pp. 33-57
[14] Y. Shi, and R.C. Eberhart, "Empirical study of particle swarm optimization", Proceedings of the 1999 Congress on Evolutionary Computation, pp. 1945 – 1950
[15] Y. Zheng, et. al., "Empirical study of particle swarm optimizer with an increasing inertia weight", Proceeding of the IEEE Congress on Evolutionary Computation, 2003
[16] Huailiang Liu, Ruijuan Su, Ying Gao , Ruoning Xu, Improved Particle Swarm Optimization Using Two Novel Parallel Inertia Weights, Second International Conference on Intelligent Computation Technology and Automation, IEEE 2009pp 185-188
[17] A. Adriansyah, and S.H.M. Amin, "Analytical and empirical study of particle swarm optimization with a sigmoid decreasing inertia weight", Regional Conference on Engineering and Science, Johor, 2006
[18] A.Nikabadi, M.Ebadzadeh , "Particle swarm optimization algorithms with adaptive Inertia Weight : A survey of the state of the art and a Novel method", IEEE journal of evolutionary computation , 2008
[19] Gao Yue-lin, Duan Yu-hong, "A New Particle Swarm Optimization Algorithm with Random Inertia Weight and Evolution Strategy" International Conference on Computational Intelligence and Security Workshops, IEEE 2007
[20] Y. Feng, G.F. Teng, A.X. Wang, and Y.M. Yao., "Chaotic Inertia Weight in Particle Swarm Optimization", In Innovative Computing, Information and Control, 2007. ICICIC'07. Second International Conference on, page 475. IEEE, 2008
[21] K. Kentzoglanakis and M. Poole., "Particle swarm optimization with an oscillating Inertia Weight", In Proceedings of the 11th Annual conference on Genetic and evolutionary computation, pages 1749–1750. ACM, 2009
[22] Y. Gao, X. An, and J. Liu., "A Particle Swarm Optimization Algorithm with Logarithm Decreasing Inertia Weight and Chaos Mutation", In Computational Intelligence and Security, 2008. CIS'08. International Conference on, volume 1,IEEE, 2008, pp. 61-65
[23] G. Chen, X. Huang, J. Jia, and Z. Min., Natural exponential Inertia Weight strategy in particle swarm optimization, In Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on, volume 1, pages 3672–3675. IEEE, 2006
[24] Hui-rong Li, Yue-lin Gao, Particle Swarm Optimization Algorithm with Adaptive Threshold Mutation, in the proceeding of International Conference on Computational Intelligence and Security, Beijing, China, December 11-December 14 2009, ISBN: 978-0-7695-3931-7,pp. 129-132
[25] H.R. Li and Y.L. Gao., Particle Swarm Optimization Algorithm with Exponent Decreasing Inertia Weight and Stochastic Mutation, In —2009 Second International Conference on Information and Computing Science,IEEE, 2009, pp 66-69
[26] Shi Y. Eberhart RC, Fuzzy adaptive particle swarm optimization, Evolutionary Computation, 2001. Proceedings of the 2001 Congress, Volume 1, Issue, 2001, pp.101 – 106
[27] J. Kennedy and R. Mendes, Population structure and particle swarm performance, in Proc. 2002 Cong. Evol. Comput., 2002, pp.1671-1675
[28] Rui Mendes, James Kennedy, and José Neves, The Fully Informed Particle Swarm: Simpler, Maybe Better, IEEE Transactions On Evolutionary Computation, Vol. 8, No. 3, June 2004
[29] Helwig S, Branke, J. ,Mostaghim, S.M., "Experimental Analysis of Bound Handling Techniques in Particle Swarm Optimization", Evolutionary Computation, IEEE Transactions on (Volume:17 , Issue: 2), April 2013, pp. 259-271
[30] Maurice clerc, Particle Swarm Optimization, ISTE (International Scientific and Technical Encyclopedia, 2006
[31] J. Kennedy and R. Mendes, Population structure and particle swarm performance, in Proc. 2002 Cong. Evolutionary Computation, 2002, pp.1671-1675

Authors' Profiles

Mrs. Snehal Kamalapur received ME (Computer Engineering) degrees from University of Pune 2003. She is working as associate professor in the K K Wagh Institute of Engineering Education and Research, University of Pune. She is currently pursuing PhD from university of Pune. She has published over 28 research papers in conferences and Journals. She is life member of various professional bodies. Her research interests include swarm intelligence, evolutionary computation and pattern recognition.

Dr. Varsha Patil is doctorate in Computer Engineering. She is working as Professor with additional reponsibility as Vice Principal in Matoshri College of Engineering & Research Centre, Nashik. She is member of board of studies of Computer Engineering at University of Pune. She has authored 2 books titled 'Discrete Mathematics' & 'Data Structures in C++' published by McGraw Hill & Oxford University Press. She has 40 papers to her credit, which are published at various national and international Journals and conferences. She is life member of various professional bodies. Her research interest includes Image processing, Evolutionary computation, Parallel Computing and Swarm Intelligence.

How to cite this paper: Snehal Mohan Kamalapur, Varsha Hemant Patil,"Adaptive Random Link PSO with Link Change Variations and Confinement Handling", International Journal of Intelligent Systems and Applications (IJISA), vol.7, no.1, pp.62-72, 2015. DOI: 10.5815/ijisa.2015.01.06

