IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 6, Ver. IV (Nov.- Dec. 2017), PP 24-31
www.iosrjournals.org

Energy minimization of protein structure using homology modeling and Particle Swarm Optimization with Dynamic Inertia Weights

Yogendra Singh Kushwah and R K Shrivastava
Department of Mathematics, Government Science College, Gwalior, MP., India.

Abstract: *Energy minimization in Bio-molecular system is an intensive application in Protein Structure Prediction (PSP). Traditional methods for reducing energy such as The Conjugate gradient method and Steepest descent method converge only local energy minima which ara highly reliant on set the parameter for calculation in initial stage. In this paper we put forward a modified PSO (DIWPSO) with control parameter D which choose best Inertia Weight strategies for energy minimization in homologous protein structure, among four best Inertia weight strategies. The proposed approach (DIWPSO) significantly improvement to search deeper energy minima and find out global energy minimum. It is found that in result analysis, proposed algorithm gives better result with regard to rate of convergence and get to minimum energy compare to traditional methods. Our homology based DIWPSO check approach utilize the design appearing to make the half of the basic masses and staying half are made haphazardly and especially handle sub-atomic compliances with a sub-atomic part reenactment bundle (AMBER12).*

Keywords: *Particle swarm optimization, Inertia weight strategies, Protein Structure, Energy Minimization, Hybridization, Convergence.*

Date of Submission: 03-08-2017 Date of acceptance: 30-12-2017

I. Introduction

Inertia weight is an crucial parameter of PSO. It has the capacity of maintain to balance between exploration and exploitation. Inertia weight is used to provide the optimum velocity to the particle at the present time stamp. When first version of PSO presented by Eberhart and Kennedy in 1995 [1], then there was no concept of Inertia Weight given by him. First time, Inertia Weight was introduced by Shi and Eberhart [2] in 1998, with constant inertia weight. They proposed to higher and small Inertia Weight which helps in global and local search. Further there are many Inertia Weight strategies proposed by researchers that give batter performance of PSO for different type of Optimization problems. A brief review of Inertia Weight strategies that is used in PSO are explained in subsequent Sections.

Bansal et al. [3] did work with different strategies of Inertia Weight. The main purpose of this studies that selecting the best inertia weight strategies. Table I shown different inertia weight strategies which are required in DIWPSO. In this paper propose a Dynamic Inertia Weight Particle Swarm Optimization Algorithm (DIWPSO) with control parameter ***D*** that choose inertia weight among four inertia weight strategies with equal probability to search those parameter of function which minimize the energy in bio-molecular system. rate of change of particle position can be updated through control parameter ***D*** which choose inertia weight from four inertia weight strategies on basis of equal probability. DIWPSO is tested over ***25*** linear, nonlinear, single variable, multivariable, Constrained and Non Constrained real world optimization problems. The results analysis shows that the modified PSO is better than basic PSO with respect to accuracy, rate of convergence and robustness.

Proteins perform all short of biological functions. These function is result of proper folded conformation adopt by proteins. Protein is a long polymer chain made by common set of ***20*** amino acids, joined by a covalent peptide bond. These protein are encode by gene in the cell system. ***3D*** conformation prediction of protein form is primary linear sequence is important in understand the biological function. The fold protein has different energy level and lower energy state fold are more stable in nature, so all protein prediction methods consider energy minimization algorithm is important for the computation protein structure prediction.

Theoretical simulations for Protein folding have lot of significant problems. It is very difficult to search the sequence of protein's amino acids in three dimensional atomic structures. It is also challenging problem that the long structure of protein molecule that has to be searched and energy minimization had to be covered. [4]. Anfinsen's hypothesis also suggested that thermodynamically structure of protein might be predicted from the sequence of amino acids by minimizing an appropriate free energy function. Even in laboratory experiments

correct folded protein is based on the minimum of total free energy has seen. Another significant problem is numerical analysis of energy optimization in energy function over native protein structures that calculate the global energy minimum has been difficult to define [5]. Therefore, this field attracted researcher to developing and optimizing simplified energy functions through parametrization [4], [6], [7], [8].

There are two Optimization method used for energy minimization. Threading [7], [9], [10], [11], [5] and the lattice models of folding [4], [12]. Another method is the decoy-based parameterization [8], in which set the parameter that search the maximizing energy gap between decoy protein structure and the native protein structure. These methods are applicable for only when the protein structure is distinct. In recently free the above condition by the method [6] , this method based on the assumption that different molecule reach to closer in native structure by set the parameter to optimize the energy function.

Above approaches to setting the parameters for energy minimization are based on experience. This parameter is chosen from favorable set of constraints for the function of energy minimization. There is no intensive work done in the field of setting the parameter to obtain the minimum energy. For this purpose propose a Dynamic Inertia Weight Particle Swarm Optimization Algorithm (DIWPSO) with control parameter D that choose inertia weight among four inertia weight strategies with equal probability to search those parameter of function which minimize the energy in bio-molecular system. DIWPSO technique reduces the hindrance to search local energy minima and find out deeper energy minima. This algorithm help to get lower energy found the global energy minima. With the help of AMBER12, molecular dynamics simulation package [13] we calculate the molecular energy of molecules during minimization.

Rest of the paper put in the following manner. In Section II describe about PSO. Various Inertia Weight Strategies for Particle Swarm Optimization are put in Section III. Implementation of DIWPSO explained in Section IV. Section V discusses the energy minimization using DIWPSO. Section VI refers to discuss about experiments and results. Finally, Section VII covered Conclusion.

II. Particle Swarm Optimization

Particle Swarm Optimization is a swarm intelligence-based optimization technique. Kennedy and Eberhart [1] was given this technique in 1995.This technique based on collective social response in a group which shown in some Insects or Animals. Such type of behavior can be observe in fish schooling and bird flocking. In PSO, every individual is known as particle that is involved in the searching of food source (optimal solution). Every particle has the capability to remember own *pbest* solution which gained during searching the *gbest* solution. Every, i^{th} particle is described as $y_i = (y_{i1}, y_{i2}, \ldots, y_{in})$. Randomly generate the population is an initial solutions which search optimal solution. The *pbest* is the personal best position of an individual in the swarm. The *gbest* is the global best position among all the individual in the swarm. The i^{th} individual best position is represented as $p_i = (p_1, p_2, \ldots, p_n)$. The velocity of a particle is the rate of change in the position and represented as $v_i = (v_{i1}, v_{i2}, \ldots, v_{in})$. The particles are update the velocity and position through following equations:

$$v_{id} = w^* v_{id} + c_1 * r_1 (p_{id} - y_{id}) + c_2 * r_2 (p_{gd} - y_{id}) \quad (1)$$

$$y_{id} = y_{id} + v_{id} \quad (2)$$

where w, c_1, c_2, r_1 and r_2 are parameters. $d = \{1, 2, 3, \ldots, N\}$, w represents the inertia weight which provides equivalence in the activity of exploration and exploitation in the search space. c_1, c_2, are acceleration constants which enforce to each individual in the swarm to attain personal best and global best position. r_1 and r_2 represent random number which get the value in the interval $[0, 1]$. If put the large value of r_1 and r_2 then particle search speed is fast, in this situation particle can miss the *gbest* position. Small value of r_1 and r_2 breakdown the speed of the particle and may be he gets pre mature solution. v is the velocity of the individual which lies between v_{min} and v_{max}. The range of the velocity is constrained of the individual in the solution space. Very small velocity might miss the opportunity of the particle to search global best solution and very high velocity might be unable to reach Optimal solution. The pseudo code of PSO algorithm is given in Algorithm (1).

Algorithm 1: Basic Particle Swarm Optimization
Input:
Initialize the parameters
Output:
(gbest) molecule
Begin:
for molecule **do**
introduce arbitrarily.

end for
iter=0
while *iter* < *Maxiter* **OR** *Minerr* condition not satisfied **do**
for each molecule **do**
Calculate wellness esteem.
if new wellness esteem is superior than *pbest* **then**
pbest = new calculated wellness esteem
end if
end for
Find *gbest* molecule from the *pbest* molecules.
for each molecule **do**
Calculate molecule speed indicated by (1).
Update molecule position concurring (2).
end for
iter = iter + 1
end while

III. Strategies Of Inertia Weight For Particle Swarm Optimization

Inertia weight is an important parameter of PSO. It has the capacity of maintain to balance between exploration and exploitation. Inertia weight is used to provide the optimum velocity to the particle at the present time stamp. When first version of PSO presented by Eberhart and Kennedy in 1995 [1], then there was no concept of Inertia Weight given by him. First time of Inertia Weight introduced by Shi and Eberhart

[2] in 1998, with constant inertia weight. They proposed to higher and small Inertia Weight which helps in global and local search. Further there are many Inertia Weight strategies proposed by researchers that give batter performance of PSO for different type of Optimization problems. A brief review of Inertia Weight strategies that is used in PSO are explained in subsequent Sections.

Bansal et al. [3] did work with different strategies of Inertia Weight. The main purpose of this study is that selecting the best inertia weight strategies. Table I shows different inertia weight strategies which are required in DIWPSO. In this paper rate of change of particle position can be updated through control

parameter D which choose inertia weight from four inertia weight strategies on basis of equal probability. DIWPSO is tested over 25 linear, nonlinear, single variable, multivariable, constrained and non constrained optimization test problems taken from literature. The results analysis shows that the modified PSO is better than basic PSO with respect to accuracy, rate of convergence and robustness. Chen et al. [14] proposed two different Inertia Weight strategies which based on Decreasing Inertia Weight. It is also analysed that proposed strategies is better than with respect to convergence of solution in early stage of approximation. It is also found that proposed strategies improve the result for continuous optimization problem.

TABLE I: Selected Inertia Weight [3]

.SN	Inertia Weight strategies	Formula of Inertia Weight
1.	Constant Inertia Weight [2]	$w = c$ $c = 0.7$ (considered for experiments)
2.	Random Inertia Weight [10]	$w = 0.5 + \text{rand}()/2$
3.	Chaotic Inertia Weight [6]	$z = 4 * z * (1 - z)$ $w = (w_1 - w_2) * \frac{MAX_{iter} - iter}{MAX_{iter}} + w_2 * z$
4.	Linear decreasing Inertia Weight [11]	$w_k = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} * k$

Fayek et al. [5] hybridize the Particle Swarm with Simulated Annealing (PSOSA) in which Inertial Weight was optimized and it was tested on urban planning problem and it found that the hybrid approach is better in respect to fast rate of convergence. It is also adaptive in urban planning problem to increased load of growing number of blocks. Chaotic Inertia Weight was proposed by Feng et al. [16]. It uses the merits of chaotic optimization. It is found that CRIW PSO performs better that RIWPSO. Malik et al. [17] proposed Sigmoid Increasing Inertia Weight strategy (SIIW) which has capability to fast convergence when fitness value of the function is minimum. In this approach Linear Increasing Inertia Weight combine with sigmoid function. To increase the overall convergence speed Gao et al. [18] proposed (LDIW), this is The Chaos mutation operator which was combined with Logarithm Decreasing Inertia Weight. The Chaos mutation increase the ability to come out of the local minima a while LDW improves its convergence speed. Exponent Decreasing Inertia Weight (EDIW) was also proposed by Gao et al. [18] that overcome premature convergence. Also stochastic mutation [19] with Exponent Decreasing Inertia Weight improves the performance of PSO and getting out from local minima.

IV. Diwpso: Dynamic Inertia Weights Based Particle Swarm Optimization

Bansal et al. [3] did work with different strategies of Inertia Weight. The main purpose of this study is that selecting the best inertia weight strategies. Table I shows different inertia weight strategies which are required in DIWPSO. In this paper, Dynamic Inertia Weight Particle Swarm Optimization Algorithm (DIWPSO) with control parameter D that choose inertia weight among four inertia weight strategies with equal probability to search those parameter of function which minimize the energy in bio-molecular system. rate of change of particle position can be updated through control parameter D which choose inertia weight from four inertia weight strategies on basis of equal probability. In the modified algorithm, the target is that the updating of molecule speed using selected inertia weight strategies presented in Table I. These strategies based on equal probability to update the molecule speed in compliance with control parameter D. The control parameter D has range in [0,1]. The algorithm (2) is the pseudo of DIWPSO.

Algorithm 2: Dynamic Inertia Weight Particle Swarm Optimization (DIWPSO)
Input:
Initialize the parameters
DIW* = [CoIW, RIW, ChIW, LDIW]
*ChIW = Chaotic Inertia Weight; RIW = Random Inertia
Weight; CoIW = Constant Inertia Weight;
LDIW = Linear decreasing Inertia Weight
w = Linear decreasing Inertia Weight
Output:
(gbest) molecule
Begin:
for individual **do**
initialize the swarm.
end for
iter=0
while *iter* < *Maxiter* **OR** *Minerr* condition not satisfied **do**
for each molecule **do**
Calculate wellness esteem.
if new wellness esteem is superior than pbest **then**
pbest = new computed wellness esteem
end if
end for
find gbest molecule from all pbest molecule.
for each molecule **do**
Choose inertia weight (w) using following strategy.
If rand() ≤ D **then**
i = **uniformly distributed random Integer in (1,4)** \\
w = **DIW [i]**
end if
Evaluate molecule speed as per (1).
Update molecule position agreeing (2).
end for
iter = iter + 1
end while

V. Energy Minimization Using Diwpso

The DIWPSO guides the demand in an educated way: exceptional parameters (to the degree satisfying lower noteworthiness scarcest) are held and misused to the most ludicrous degree through time, while new scopes of the parameter space are investigated intentionally through blend and change. The focal accepted is to encode the preoccupation parameters and conditions into strings, and apply the DIWPSO number to the strings with a target work mirroring the essentialness of the structure noteworthiness. We apply DIWPSO with a specific extreme goal to look for parameters that purpose of control the free vitality of a bio-sub-atomic framework. A subset of these parameters is generally chosen for progress and is encoded into DIWPSO strings. The strategy of parameters to be advanced is issue subordinate, and is picked in context of the physical prerequisites and setups of the structure. It is precious watching that a solitary vitality minimization figuring in AMBER12 has five particular stages: The hidden four phases are atomic segments eras, and the last stage is the

noteworthiness minimization step utilizing steepest jump or conjugate inclination technique. AMBER12 parameters contain a few classes concerning limit conditions, requirements, potential vitality limits, purpose of merging of mass improvements, non-reinforced affiliations, and program control parameters for these calculations. These five phases reflect a technique for instatement, warming, dependable temperature sub-atomic stream spread, cooling, and vitality minimization. We utilize the AMBER12 bundle [13] to enlist the vitality of bio-atomic structures.

Two delineation methods that are accessible for the bio sub atomic framework, i.e. In the present study we utilized true blue respect encoding course of action for the parameters. Space settled Cartesian direction [22] and inside headings framework. A significant piece of the reviews have demonstrated that space-settled Cartesian heading delineation enhances the feasibility of the consider separated and interior supports portrayal, along these lines space-settled. The space settled portrayal required *3n* factors while inward deals with technique required *n(n - 1)=2* parts. Cartesian headings delineation is utilized.

The arrangement of directions are spoken to as x = (x1,y1,z1, x2, y2, z2, ...) and V (x) is the potential vitality, then the goal is to discover x that limit the V (x).

$$FitnessFunction = MIN\ (V\ (x)) \quad (3)$$

Figure 1 describes the flow of process used. The half of the initial population is getting from PDB-Database using PSIBLAST and second half initialize randomly. After initializing the population, fitness is evaluated and then, update or generate the new population according to DIWPSO till the termination criteria is met.

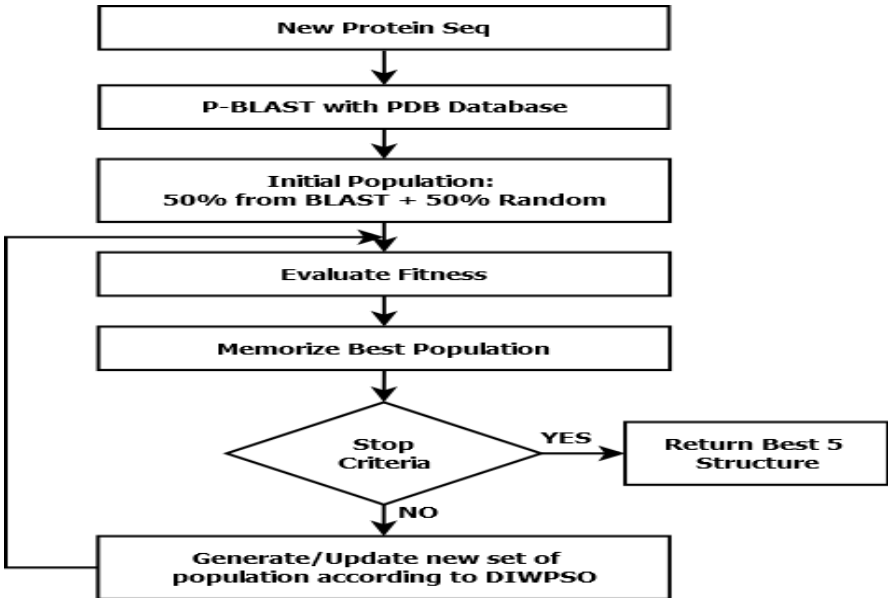

Fig. 1: Energy minimization methodology using DIWPSO.

The initial half population is getting from PSI-BLAST and the half population (dihedral angles (ϕ , ψ)) is generated randomly, with each sampled uniformly between [-180^0 , 180^0]. The conformation is considered as α-helical when the dihedral angles (ϕ , ψ) fall in the range [-60 ± 45^0 , -50 ± 45^0] [23]. e.g. a four nucleotides protein is represented by real string of length 12.

x1	y1	z1	x2	y2	z2	x3	y3	z3	x4	y4	z4
0.47	3.35	4.16	-2.32	-4.17	-5.64	-1.86	-3.89	-1.23	5.79	2.49	3.66

There are two cases of termination condition are apply in DIWPSO. The first condition based on the convergence on generation that is when The desired generation turnout, the algorithm terminated. The second condition apply when the best fitness value get in the population (referred to as convergence on global best).The algorithm terminates If the ratio between the best value of the N^{th} previous generation, where N is specified by the user, and the best value of the current generation is larger than a specified value α $(0 < \alpha \leq 1)$.

VI. Results Analysis And Discussion

This section discusses the environment setup, parameter setting, structure and minimum energy and cross validation of results. DIWPSO algorithm apply on the molecular system which consists of five amino acids Tyrocine, Veline, Glutemine, Lysine and Arginine shown in Figure 2 for the chemical structure), one sodium ion (Na+), three chlorine ions (CL^-), and water as the solvent with 921 water molecules placed in a periodic box with an initial dimension of 3nm on each side. The results are compared with the Genetic Algorithm, Particle Swarm
Optimization (PSO), Ant Colony Optimization (ACO) and Classical Differential Evolution (DE) algorithms.

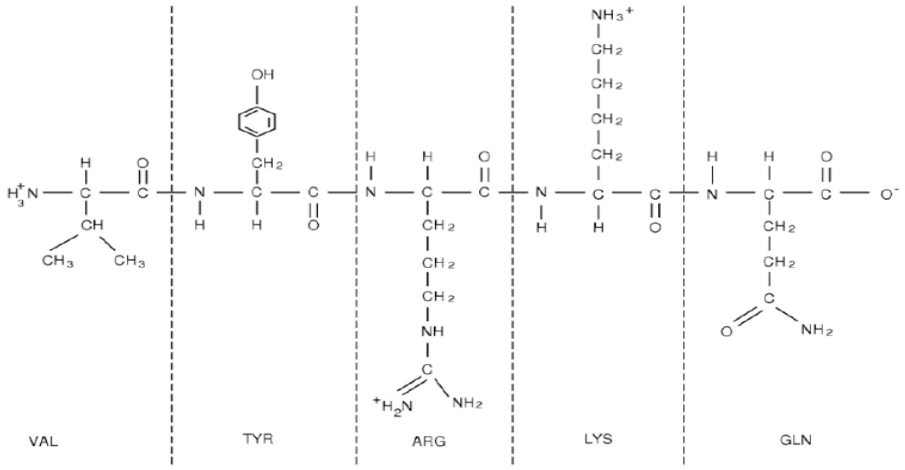

Fig. 2: Simplified diagram of test molecule consisting of five amino acids: H-VAL-TYR-ARG-LYS-GLN-O.

The motive of the proposed approach is to set the parameter in the string for energy minimization in protein structure prediction. Molecule position in a protein fold encode in string through set the optimize parameter. For calculation of minimum energy we compute its fitness value by AMBER12. We are choosing 38 parameters for energy minimization in this experiment. Also include temperature values, the number of time steps computed in each AMBER12 stage. Maxwell distribution is used for sampling the initial atomic velocities at the particular temperature. In a real value string encode these parameters with length of 114 (3 * 38).

A. Environment setup

For energy computation The AMBER12 [13] package is used, with the parameter setting as imin=1, maxcyc=200, ncyc=75, ntpr=10, ntb=0, igb=0, cut=12. Implementation of All the algorithms in C++ in Linux environment and executed on Intel core i5 with 3.2 GHz processor with 4GB of RAM.

B. Required Parameter

Important task is to set the Parameter for algorithms. Taken 50 population size for initial approximation and it can be iterating up to 12000 generation for each algorithms. *3n;* is the dimension where *n* is number nucleotides. Table II summarizes the default parameter setting for each algorithm.

TABLE II: Parameter settings for energy minimization

.SN	Algorithm	Parameters
1.	GA	CR=0.8, MR = 0.2
2.	DE	CR=0.8, F = 0.6
3.	PSO	C1, C2 =1.5, w = 0.7
4.	ABC	NP=30, FOOD SOURCE=NP/2
5.	DIWPSO	C1, C2 =1.5, D = 0.7

C. Minimum energy

All the algorithm are run for 1400 iterations with initial population size of 50. The Figure 3 shows the final minimized energy for the targeted protein sequence discuss earlier. The minimum energy of $-32{:}65*10^4$ *KJ/mole* from DIWPSO and the maximum energy is getting through DE $-30{:}90*10^4$ *KJ/mole*.

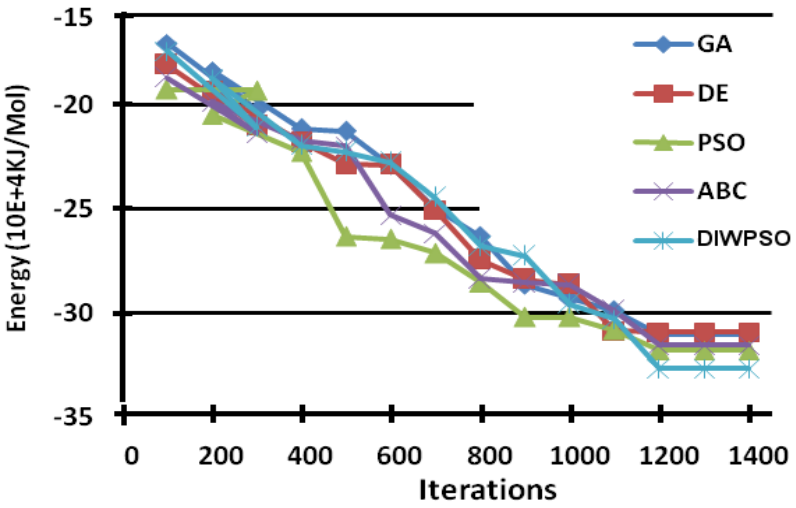

Fig. 3: Minimum energy getting from different algorithms.

The Table III shows the minimum energy getting through DIWPSO for the public available PDBs. The performance of DIWPSO is better in 50% cases.

TABLE III: Minimum Energy Comparison of DIWPSO with publicly available protein.

PDB ID	Actual Energy (KJ/mole)	Energy getting through DIWPSO (KJ/mole)
1FTJ	-104132.86	-109258.56
1II5	-133269.75	-116491.57
1PB7	-100296.00	-100352.89
1S7Y	-115314.22	-116170.86
1S9T	-136848.94	-124180.05
2FTJ	-133161.39	-136154.82
2II5	-192040.50	-164142.68
2PB7	-178509.10	-156986.05
2S7Y	-123616.28	-124646.96
2S9T	-178026.36	-127477.69

VII. Conclusion

Modified PSO(DIWPSO) presented in This paper. DIWPSO performed with control parameter D. D select Inertia Weight with equal probabilities among four Inertia Weight strategies in each approximation. We have performed the minimization of total empirical energy of protein structure using DIWPSO. We found that DIWPSO outperform in energy minimization of the targeted protein sequence over GA, DE, PSO and ABC algorithms. The result are also compared with public available PDBs and it is found that DIWPSO perform better in 50% cases. The advance Technique of the algorithms is the future scope for big size protein structure; it is also believe that by merging more effective inertia weight strategy may given more better results with higher efficiency.

References

[1]. R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In Micro Machine and Human Science, 1995. MHS'95.,Proceedings of the Sixth International Symposium on, pages 39–43. Ieee, 1995.

[2]. Y. Shi and R. Eberhart. A modified particle swarm optimizer. In Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on, pages 69–73. IEEE, 1998.

[3]. JC Bansal, PK Singh, Mukesh Saraswat, Abhishek Verma, Shimpi Singh Jadon, and Ajith Abraham. Inertia weight strategies in particle swarm optimization. In Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on, pages 633–640. IEEE, 2011.

[4]. M. H. Hao and H. A. Scheraga. How optimization of potential functions affects protein folding. Proceedings of the National Academy of Sciences, National Academy of Sciences Publication, 93(10):4984– 4989, 1996.

[5]. K. Koretke, Z. Luthey, and P. G. Wolynes. Self-consistently optimized energy functions for protein structure prediction by molecular dynamics. Proceedings of the National Academy of Sciences, National Academy of Sciences Publication, 95(6):2932–2937, 1998.

[6]. J. B. Rosen, A. T. Phillips, S. Y. Oh, and K. A. Dill. A method for parameter optimization in computational biology. Biophysical Journal, Elsevier, 79(6):2818–2824, 2000.

[7]. R. A. Goldstein, Z. A. Luthey, and P. G. Wolynes. Protein tertiary structure recognition using optimized Hamiltonians with local interactions. . Proceedings of the National Academy of Sciences, National Academy of Sciences Publication, 89(19):9029–9033, 1992.

[8]. C. Seok, J. B. Rosen, J. D. Chodera, and K. A. Dill. MOPED: Method for optimizing physical energy parameters using decoys. Journal of Computational Chemistry, Wiley, 24(1):89–97, 2003.

[9]. G. M. Crippen and V. N. Maiorov. How many protein folding motifs are there? Journal of Molecular Biology, Elsevier, 252(1):144–151, 1995.
[10]. P. D. Thomas and K. A. Dill. An iterative method for extracting energylike quantities from protein structures. Proceedings of the National Academy of Sciences, National Academy of Sciences Publication, 93(21):11628–11633, 1996.
[11]. L. A. Mirny and E. I. Shakhnovich. How to derive a protein folding potential? A new approach to an old problem. Journal of Molecular Biology, Elsevier, 264(5):1164–1179, 1996.
[12]. I. Shrivastava, S. Vishveshwara, M. Cieplak, A. Maritan, and J. R. Banavar. Lattice model for rapidly folding protein-like heteropolymers. Proceedings of the National Academy of Sciences, National Academy of Sciences Publication, 92(20):9206–9209, 1995.
[13]. A. W. G¨otz, M. J. Williamson, D. Xu, D. Poole, S. Le Grand, and R. C. Walker. Routine microsecond molecular dynamics simulations with AMBER on GPUs. Journal of Chemical Theory and Computation, ACS Publications, 8(5):1542, 2012.
[14]. 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.
[15]. W Al-Hassan, MB Fayek, and SI Shaheen. Psosa: An optimized particle swarm technique for solving the urban planning problem. In Computer Engineering and Systems, The 2006 International Conference on, pages 401–405. IEEE, 2006.
[16]. 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.
[17]. R.F. Malik, T.A. Rahman, S.Z.M. Hashim, and R. Ngah. New Particle Swarm Optimizer with Sigmoid Increasing Inertia Weight. International Journal of Computer Science and Security (IJCSS), 1(2):35, 2007.
[18]. 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, pages 61–65. IEEE, 2008.
[19]. 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, pages 66–69. IEEE, 2009.
[20]. R.C. Eberhart and Y. Shi. Tracking and optimizing dynamic systems with particle swarms. In Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, volume 1, pages 94–100. IEEE, 2002.
[21]. J. Xin, G. Chen, and Y. Hai. A Particle Swarm Optimizer with Multistage Linearly-Decreasing Inertia Weight. In Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on, volume 1, pages 505–508. IEEE, 2009.
[22]. J. A. Niesse and H. R. Mayne. Global geometry optimization of atomic clusters using a modified genetic algorithm in space-fixed coordinates. The Journal of Chemical Physics, AIP Publications, 105:47–51, 1996.
[23]. Y. Okamoto, M. Fukugita, T. Nakazawa, and H. Kawai. _-Helix folding by Monte Carlo simulated annealing in isolated C-peptide of ribonuclease A. Protein Engineering, Oxford University Press, 4(6):639–647, 1991.

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

Yogendra Singh Kushwah "Energy minimization of protein structure using homology modeling and Particle Swarm Optimization with Dynamic Inertia Weights." IOSR Journal of Computer Engineering (IOSR-JCE) 19.6 (2017): 24-31.