Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 46 No. 04 (2025)

Differential Evolution Based Particle Swarm Optimization Algorithm for Protein Ligand Docking

[1]*Dr.P. Lakshmi, [2]E. Pavithra Vishalini, [3]Dr. D. Ramya Chitra,

[1]Assistant Professor, Department of Computer Science, PSG College of Arts and Science, Coimbatore, Tamilnadu.

[2]Assistant Professor, Department of Computer Science, Pollachi College of Arts and Science, Pollachi, Coimbatore,

[3]Assistant Professor, Department of Computer Science, Bharathiar University, Coimbatore, Tamilnadu.

Abstract: Molecular Docking is a computerized tool used in discovering a drug. Molecular Docking (MD) which helps to predict the small molecule ligand is able to bind to disease target protein. In MD a configuration molecular structure is generated by the conformational search algorithm and then the position is evaluated based on fitness function. Here fitness value is least binding energy during the interaction of protein and Ligand. The more the least binding energy, the more the ligand is stable in the complex. In this research a differential evolution-based particle swarm optimization algorithm is proposed as a search algorithm for conformational space in protein ligand docking. Using a dataset of 1089 bimolecular complexes from PDBbind the lowest binding energy and time efficiency were tested. The proposed algorithm demonstrates superior when tested with six other existing algorithms.

Keywords: Molecular Docking, Search algorithm, Evolutionary algorithms, Particle Swarm Optimization, Binding Affinity.

1. Introduction

Molecular Docking is the computational molecular modeling drug design techniques which plays an indispensable role in the process of drug design (NS Pagadala et al, 2017). Pharmaceutical companies utilize the computational techniques (L Pinzi et al, 2019) at various stages in the drug design process for successful and profitable factor. Docking is also a computational method using to determine the binding affinity and the orientation of the complex in the active site of the target protein. There are various computational tools available for protein ligand docking. There are two elements used in Docking which are search algorithms and scoring function. Process and Example of Molecular Docking are shown in Fig 1 and Fig 2 respectively

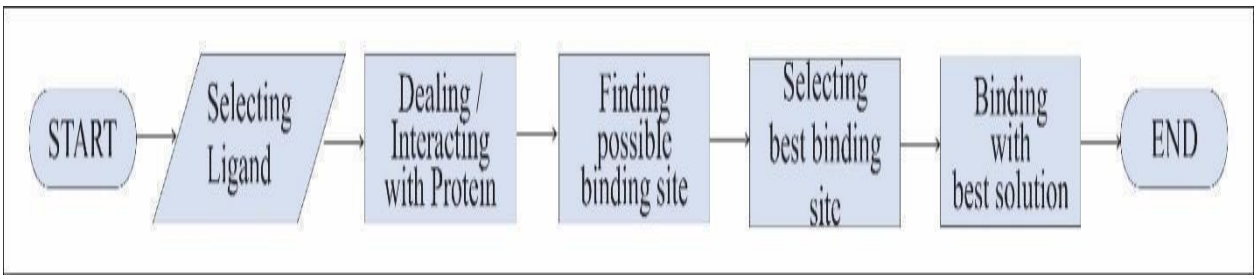


Figure 1 Process of Molecular Docking

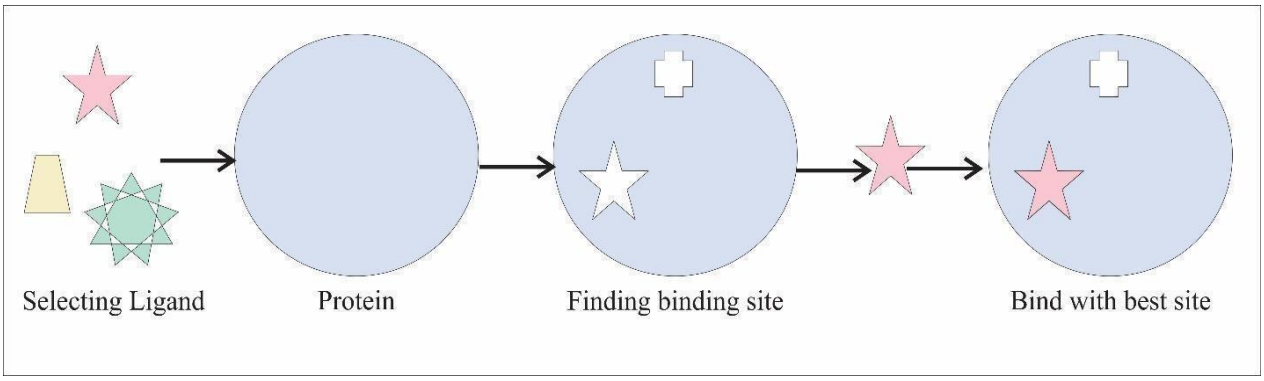


Figure 2 Example for Molecular Docking

In Molecular Docking there are several processes that consist of (i) ligand is selected to bind to target protein, (ii) to find the binding sites that are available in receptor protein using search algorithm and scoring function (iii)obtain best binding site and binding with optimal position technically. This technical process is called Docking

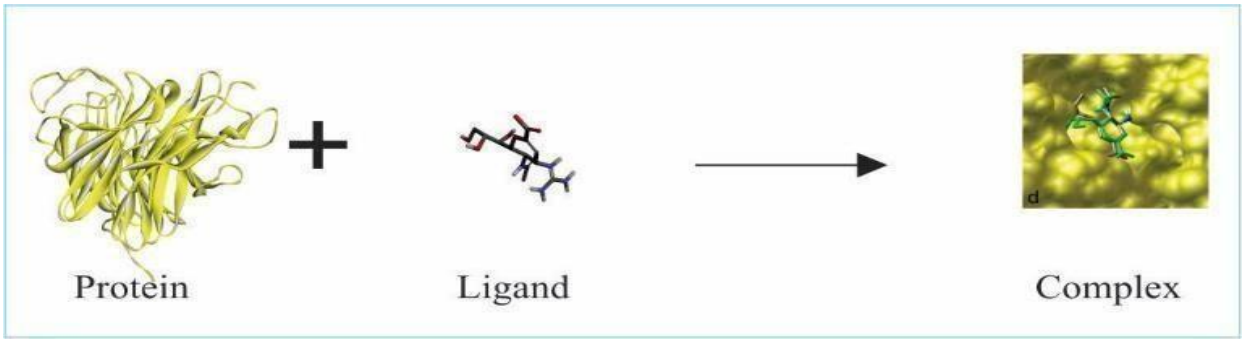


Figure 3 Molecular Docking workflows

The Search algorithm generates the molecular conformation (position of the ligand docked in the active site) and then the fitness value is evaluated for such conformation. The best fitness value is evaluated using scoring functions. AutoDock4 is a docking open-source tool developed at The Scripps Research Institute. The search algorithms used in AutoDock4 include genetic algorithm (C. M. Oschiro et al, 1995), simulated annealing, and hybrid local search GA (GM Morris et al , 1998). In this research a hybrid differential evolutionary (DE) based particle swarm optimization (PSO) algorithm is proposed. The environment and scoring function are evaluated using AutoDock4.2. The performance is compared with the algorithm Particle Swarm Optimization PSO, Lamarckian Genetic Algorithm LGA, Ant Colony Optimization ACO, Differential Evolution DE (R Thomson et al, 2003) , Monte Carlo Simulated Annealing SA (W Forli et al 2012), Artificial Bee Colony ABC(Uehara, S et al , 2015)and proposed Differential Evolution based Particle Swarm Optimization PSODE algorithm. And the results clearly show that the proposed PSODE works prior to all those algorithms tested in terms of accuracy. The remaining sections of this paper are organized as follows. Section 2 defines the materials and method of the projected system. Section 3 illustrates the experimental results and discussion on various datasets and finally Section 4 concludes the paper with future enhancements.

2. Materials And Methods Artificial Bee Colony (ABC)

The ABC algorithm is a swarm-based, metaheuristic algorithm depends on s the searching performance of honey bee

groups or colony (Dervis Karaboga et al 2008). The classical technique is built using three major elements:

a) Employed

b) Unemployed foragers

c) Food sources.

In ABC algorithm the first part consists of employed bee whereas the second art onlooker bee. Each employed bee searches a food source, hence employed bee is equivalent to numeral count of food source. Employed bee finds the possible food source and share food source information to onlookers by dancing (Pratyusha et al,2013). The onlookers detect the employed bees dance inside the hive, and to choose the food source, however scouts search arbitrarily for new food sources (Uehara, S et al, 2015).

The search sequence of ABC involves rules such as:

(i) Employed bees search for food sources and shares details about food source to onlooker bees;

(ii) Onlooker bees computing the nectar quality and selecting the food sources after gaining information about employing bees

(iii) Defining the scout which bee transferring employee bee to feasible food sources.

In ABC, a group of artificial bee examines optimal food sources (quality solutions based on fitness value). In molecular docking ABC finds the location of a protein ligand binding place (food source) characterizes a solution vector of the optimizing problem, and feature of the binding area (nectar amount) is signified by a fitness value calculated. The three kinds of bee survey for a global best solution in D-dimensional real grid area (parameter space), where D matches to the number of optimization structures (translation, orientation and conformation of ligands for protein-ligand docking)

Algorithm 1 Pseudocode of ABC algorithm for Docking

1. Initialize the population $x\,i\,(i = 1, 2, \ldots, SN)$
2. Calculate the fitness energy F(v) of binding position using eq. $E_{bind} = E_{inter} + E_{intra}$
3. Repeat
4. For each employed bee Phase
   Select random dimension of ligand as solution v_i
   Calculate its fitness value F (v_i) using eq. $E_{bind} = E_{inter} + E_{intra}$
   Calculate the binding area values Pi for the solution (v_i)
5. For each onlooker bee Phase
   Select a position x_i depending on P_i
6. Scout Phase
   If limit reached reinitialize the position
   Evaluate evolutionary rate
7. Update x_0 best position
8. Return x_0

Particle Swarm Optimization (PSO)

The Particle Swarm Optimization (PSO) (Kennedy et al, 1995) introduced an optimized search method which is an evolutionary algorithm based on population at every iteration, all particles move in the problem space (search area) to find the optimal global solution. Every particle has a current position vector and velocity vector for further movement. In each iteration method, each solution is nominated by the fitness function being optimized. Every particle saves its position, moves in the search area randomly and calculate its fitness along with its velocity (Kai Chen et al, 2006).

Additionally, it concerns the best fitness value, that have been expert during the operation of the algorithm the solution that attained this fitness, is mentioned as the individual best position. Liu et al, 2013 described it as randomized, population-based optimization method which was moved by the flocking performance of birds and human.

PSO have been efficiently used in numerous real-world problems. The main area of such application is frequent purpose optimization, PSO is in standard well appropriate method to the docking calculation were a real- valued illustration of the ligand have to be reduced with respect to the fitness function. In PSO particles is used as population of solutions. Such particle moves over the search space to predict the best position (Hung-Ming Chen et al, 2007). The motion of a particle is inclined by the particle's local search history and by the best positions that have been found by other neighbored particles in the swarm. For gbest PSO method, the velocity of particle i is calculated by

$$v^{t+1} = v^{t} + c\, r^{t} [P - x^{t}] + c\, r^{t} [G - x^{t}] \quad \text{----------(1)}$$

v_{ij}^{t} is the velocity vector of particle i in dimension j at time t ;

x_{ij}^{t} is the position vector of particle i in dimension at time t ;

$P_{best,i}$ is the personal best position of particle i in dimension j found from initialization through time t;

G_{best} is the global best position of particle i in dimension j found from initialization through time t;

c_1 and c_2 are positive acceleration constants which are used to level the contribution of the cognitive and social components respectively;

r_1 and r_2 are random numbers from uniform distribution at time t.

For each particle has its current i position on searching space and assume its optimal position as p_i, (Nama sivayam, 2007). The particle movement is evaluated by the velocity vector v_i. For every iteration particle moves hence current position I and personal best position p_i is calculated based on the velocity v_i if it is better than the Pbest position $f(x_i) < f(p_i)$, then $x_i$updated as new best position, i.e., $p_i = x_i$. Then it randomly chose another particle n within the search and v_i velocity is calculated in every dimension d. The process is repeated until reaches terminal condition. often chosen as the particle that has the best personal best position within the search area (Marcus et al, 2015).

Algorithm 2 Pseudocode of PSO algorithm

Initialize protein ligand binding position x_{ij}
Initialize confidence of position c_1, c_2, velocity v_{ij} is the speed of changing position particle and r_1 and r_2 are vector (0,1).
Evaluate fitness value F(x) is the binding energy using eq. $E_{bind} = E_{inter} + E_{intra}$
Calculate Pbest value using equation (3.2)
while
    number of iteration condition fail do
    calculate velocity and orientation of ligand i
    Evaluate fitness value for new position F(x)
    If $F(x_i) \leq (x_{Pbest,\,i})$ /* *P_{best} is the best local new position of ligand* */
    then $x_{Pbest,\,i} = x_i$
    end if
    If $F(x_i) \leq F(x_{gbest,\,i})$ /* *G_{best} is the global best position of ligand* */
    then $x_{gbest,\,i} = x_i$
    end if
End while

Differential Evaluation (DE)

The Differential Evolution is a optimizing algorithm based on population search method to evaluate the global optimal solution for optimization problems in an efficient way. Population size P is initialized with the parameter value, for every population dimensional vector parameter is assumed as an individual population in DE procedure (Rene Thomsan,2003).t is the generation evaluate at every iteration, a maximum number of generations (tmax) is to obtain a global best solution. x_i, tis the t^{th} generation of i^{th} population is write as {x1,i,t , x2,i,t, . . . , xD,i,t}(Vitaliy , 2006).

Mutation

Next to initialization process of population, Differential Evolution chooses a donor vector which is called mutant vector, $V_{i,\,t}$ equivalent to every individual vector $x_{i,\,t}$ in the current position through mutation. The i^{th} the donor vector of the current generation, t is denoted as: $V_{i,t}$ = {v1,i,t , v2,i,t, . . . , vD,i,t}. Zhenyu et al , 2007 listed the regularly used mutation strategies.

Crossover

In Crossover, Phase trial vector $u_{i,\,t}$ is generated for every pair of mutant and target vector by performing the crossover operation $u_{i,\,t}$ = {u1,i,t , u2,i,t, . . . , uD,i,t}. In DE algorithm crossover operator plays an vital one to detect the search space. Generally, there are two types of crossover methods used exponential and binomial (Daniela Zaharie, 2009). Exponential crossover, choose an initial point integer n ∈ [1, D] randomly, in the vector where the interchange of components starts with primary vector. Then the trial vector is calculated in exponential crossover. In binomial crossover, crossover performed based on the ratio (CR) at every dimension (D). While the random number ran (0, 1) ∈ [0, 1], and crossover ratio ∈ [0, 1] is merely taken from the equivalent target vector $x_{i,\,t}$. Then the trial vector is calculated in binomial crossover.

Selection

In this phase selection process is performed to reinitializing parameter with random and uniform with selection operator predefined range operator (Wenyin Gong et al, 2011). The selection operator decides the target vector x_i, t or the trial vector u_j, i, t may continue on further generation by considering its fitness values. The better individual can be selected as: $x_{i,\ t+1} = u_{i,\ t}$ if $f(u_i, t) \leq f(x_i, t)$ x_i, these steps are repeatedly iterated into subsequent DE generations till reach the limit or terminal condition (Yang,2007).

Algorithm 3 Pseudocode of DE algorithm.

Initialize the current position
For t=1 to N do
/ * mutation*/
For each position x_i pick three randomlydifferent orientation x_1, x_2, x_3
Compute the new position using eq. $v_i^{(t+1)} = x_{r1}^{(t)} + F(x_{r2}^{(t)} - x_{r3}^{(t)})$
Evaluate objective function $F(x_m)$ using eq. $E_{bind} = E_{inter} + E_{intra}$
/ * Crossover*/
Crossing the mutant dimension i , j individual randomly
Evaluate objective function $F(x_c)$ using eq. $E_{bind} = E_{inter} + E_{intra}$
/ End of Crossover /
/ * Selection*/
If $F(x_m) < F(x_c)$
$F(xi) = F(x_m)$
Else
$F(x_i) = F(x_c)$
/ End of Selection /
Return $F(x_i)$

ALGORITHM 4 Pseudocode of DE algorithm for Docking

INITIALIZE the initial binding population P (x);
EVALUATE the binding energy at each position in P (x);
While(termination condition meets not true) do
Begin
x=x+1;
/* Mutation */
Mutation performs at every binding position in M (t) and moves to the resulting position
corresponds to the probability P_m
/* Crossover */
Select two positions and perform crossover with probability P until it reach the limit L(t).
/*Lamarckian Evolution */
Perform displacement on each position in L(t) and promote the population C (t);
EVALUATE: The energy at every position in C (t);
End

Lamarckian genetic algorithm

LGA is the local search hybrid Genetic algorithm. At LGA, every state variable corresponds to gene. Ligand coordinates are considered as genotype and atomic coordinate in docking space are described as phenotype. Then the nergy is calculated from the genotype form its phenotype. In LGA the genotype variables are not inverted back .[12]

Differential Evolution based PSO (PSODE)

In the PSODE algorithm, PSO is executed at the beginning of each generation. In Molecular Docking PSO algorithm works as searching algorithm to find the different position of the binding site. PSO uses particle and swarm variables to execute. Here swarm represent the population or possible space in the binding area and particle represent the candidate in the binding area. The Fitness function is the lowest energy produced when binding which is calculated at each position by using chemical calculation formula. The global best solution is calculated to produce a optimal solution. After execution of PSO, process switches to DE. Results of PSO and DE will be compared and the best one will be updated. The process executes based on the probability of generation.

$$f*(t) = f*(t+1) = f*(t+2) = \blacksquare\blacksquare\blacksquare = f*(t+N) \quad (2)$$

Algorithm 5 Pseudo code of PSODE algorithm for Docking.

Initialize protein ligand binding position x_{ij} confidence of position c_1, c_2, velocity $v_{ij}$is the speed of changing position particle and r_1 and r_2 are vector (0,1).

Evaluate fitness value F(x) is the binding energy using eq. $E_{bind} = E_{inter} + E_{intra}$ Calculate Pbest value using equation

while

number of iteration condition fail do calculate velocityand orientation of ligand i Evaluate fitness value for new position $F(x_i)$

If $F(x_i) \leq F(x_{Pbest, i})$ /* *Pbestis the best local new position of ligand* */ then $x_{Pbest, i} = x_i$

end if

If $F(x_i) \leq F(x_{gbest, i})$ /* G*bestis the global best position of ligand* */ then $x_{gbest, i} = x_i$

end if

Update i^t particle position and velocity x_i and v_i Update fitness F(x)

For t=1 to N do

/ * mutation*/

For each position x_i pick three randomlydifferent orientation x1, x2, x3 Compute the new position using eq.$v_i(t+1) = xr^1(t) + F(xr^2(t) - xr^3(t))$ Evaluate objective function F(mx) using eq. Ebind = Einter + Eintra

/ * Crossover*/

Crossing the mutant dimension i , j individual randomly

Evaluate objective function F(cx) using eq. Ebind = Einter + Eintra

/ End of Crossover /

/ * Selection*/ If F(mx) < F(cx)

F(xi) = F(mx) Else

F(xi) = F(cx)

/ End of Selection / If F(xi) < F(x) Update F(xi) = F(x) End while

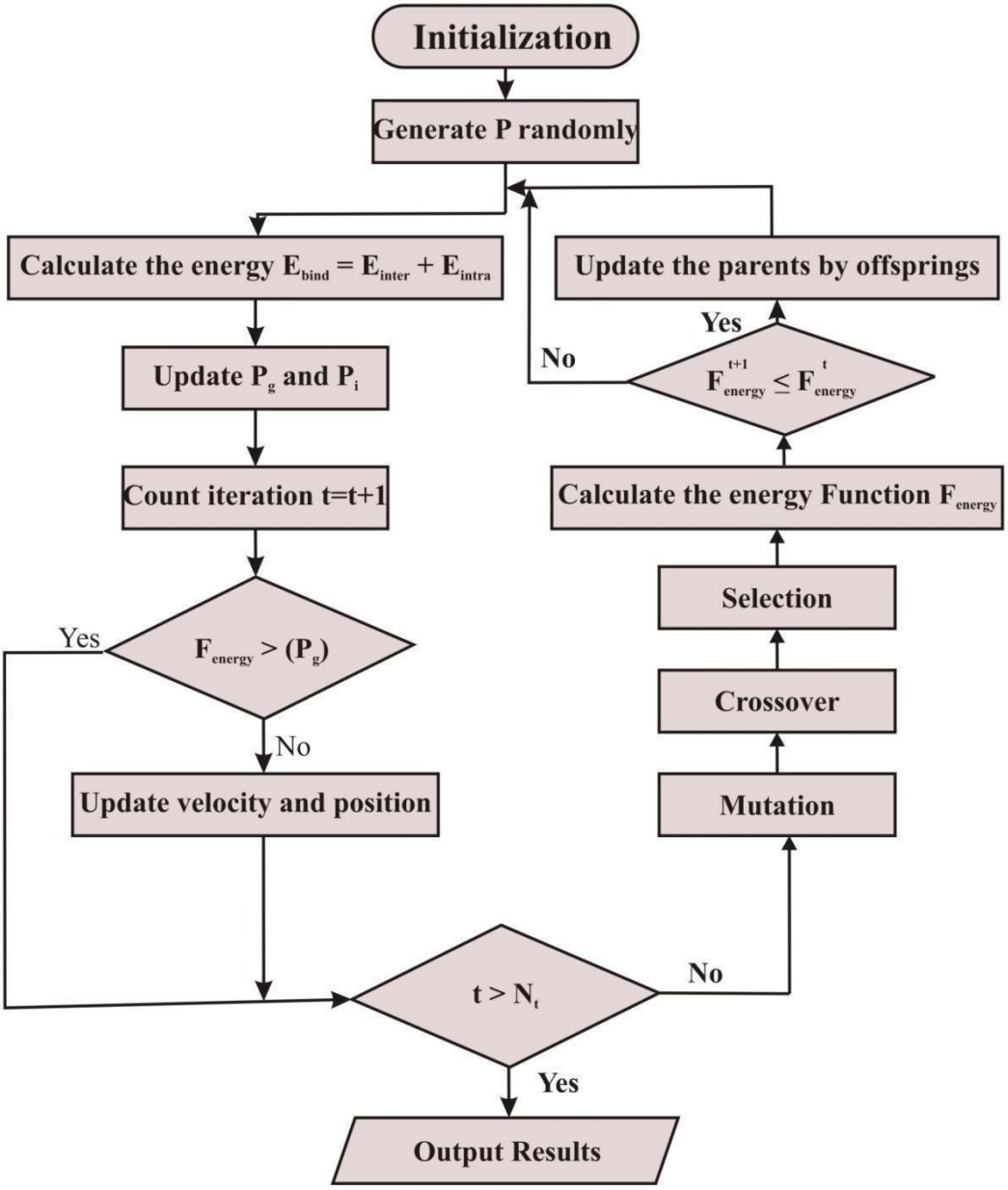


Figure.4 The workflow of proposed hybrid PSO and DE algorithm.

Figure 4 represents the framework between the PSO and DE, named PSODE is finding the optimal solution for the docking bind problem. In PSODE the velocity of particles represents the position of ligand particle. In DE the particle velocity is updated through the selective processes using following Equation

$v_i(t+1) = v_i(t) + (u_i(t) - x_i(t))$ (3)

r and(0, 1), where vi (t) is the velocity of I th particle, ui (t) represents the velocity of trial vector , and xi (t) is the velocity of the target vector at generation t. The proposed algorithm experiments on 100 complex structures. The performances are calculated based on the accuracy, sensitivity, sensitivity, and Root Mean Square Deviation. The experimental results confirmed that the PSODE algorithm performs every protein ligand complex and improves the performance in a efficient way. In this chapter, the system architecture of the proposed algorithm is discussed. The

existing and proposed algorithms are discussed in this algorithm. Finally, the proposed algorithm flowchart and advantages are described in this chapter.

The Dataset used for experimental study

In this research, protein-ligand complexes randomly selected from a SB2012 complex database that is used for the computational experiments.

Data preparation

Protein and ligand structures are pre-processed using AutoDock4.2.6. Pre-processing steps include removing water molecules, adding hydrogen bond for predicting better results, then the file converted to pdbqt required format, allocating the grid parameter file .gpf (searching space) for docking. Using Auto Grid binding energy is calculated with default grid size 22.5 ˚A.

Parameterization

The initial population used for PSO, DE, ABC and PSODE algorithms was 60; the energy function evaluations was set to maximum number of 22,000 generations. The possibility of executing a local search on an individual was

0.05. The other parameters provided by the default setting were the same as in Auto Dock.

Software

Auto Dock is an open-source molecular docking software for protein-ligand docking, which predicts how ligands bind to the target or the grid area on the protein. A grid parameter file .gpf consist the information of searching space area of the docking. Autodock4.2.6 environment is used for docking calculation. Python 2.7 programming language was used for developing the code for searching algorithm.

3. Results And Discussion

Performance measures used in this study

The comparison is made in terms of the performance metrics referred to as the accuracy, specificity, sensitivity, and F1-score that are defined in the following subsections.

Sensitivity

Sensitivity or recall is the percentage of positive circumstances that were correctly recognized, as calculated using the equation,

$$sensitivity = \frac{True\ Positive}{\quad} \quad (4)$$

$$True\ Positive + False\ Negative$$

In Figure 5 comparison results of the proposed approach with the existing method in terms of Sensitivity. Datasets are represented in X-axis and sensitivity values are denoted in Y-axis. From the bar chart, it is increased for the proposed approach compared to the existing approach.

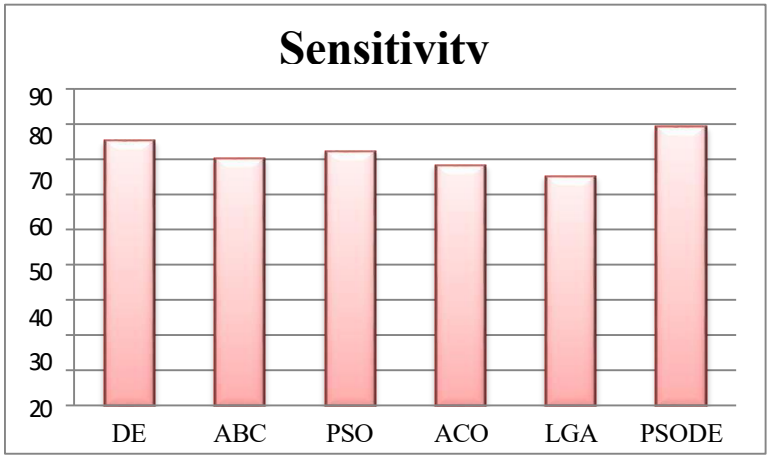


Figure 5 Performance Comparison of existing and proposed approaches for Sensitivity.

Specificity

True Negative rate (TN rate), or specificity, is the proportion of actual negatives which are predicted to be negative and is calculated as follows,

$$Specificity = \frac{True\ Negative}{True\ Negative + False\ Positive} \quad (5)$$

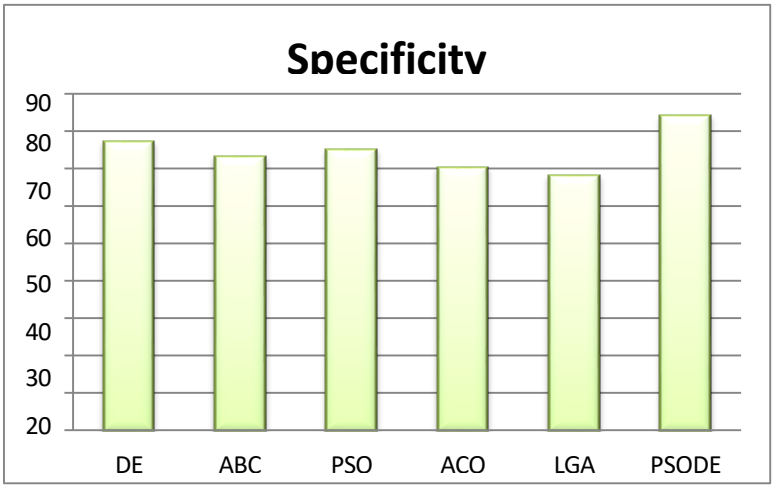


Figure 6 Performance Comparison of existing and proposed approaches for Specificity

In Figure 6 Comparison results of the proposed approach with the existing method in terms of Specificity. Datasets are represented in X-axis and specificity values are denoted in Y-axis. From the bar chart, the proposed approach provides high specificity.

Accuracy

Accuracy is defined as the overall accuracy rate or classification accuracyand is calculated as follows,

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \text{----------(6)}$$

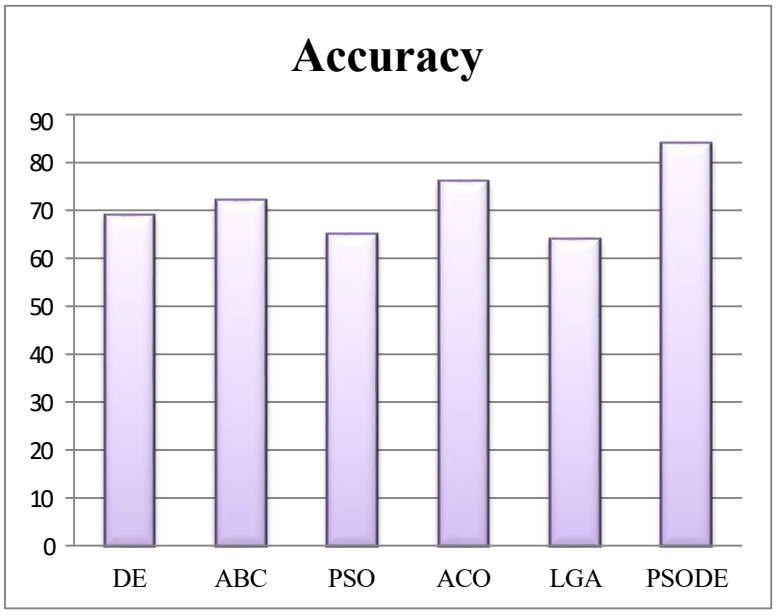


Figure 7 Performance Comparison of existing and proposed approaches for Accuracy

Figure 7 shows that the comparison of the proposed approach and existing method in terms of Accuracy. Datasets are represented in X-axis and accuracy values are denoted in Y-axis. The accuracy value is increased for the proposed approach compared to the existing approach.

Comparison of Performance of existing and proposed methods for protein-ligand complex data sets

Table 1displays the comparison of the lowest docking energy of DE, PSO, ABC and PSODE 12 out of 100 complex ABC shows the best least energy whereas hybrid PSODE presents 88 out of 100 complexes. This shows the effect hybrid algorithm. In contrast the RMSD analysis of PSODE shows better results than ABC, DE, and PSO.

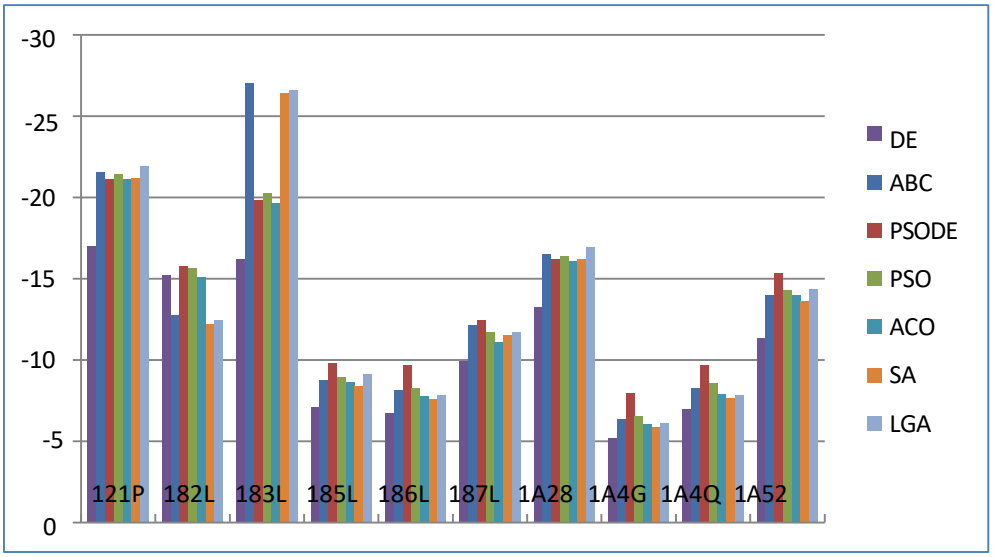


Figure 8 Protein ligand complex dataset with comparison of algorithms from 121P to 1A52

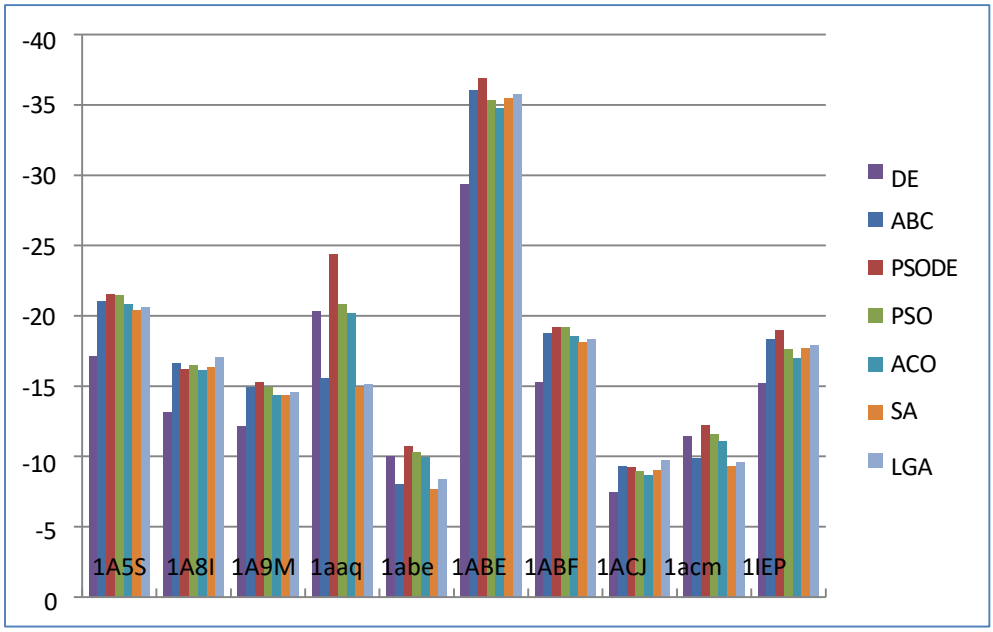


Figure 9 Protein ligand complex dataset with comparison of algorithms from 1A5S to 11EP

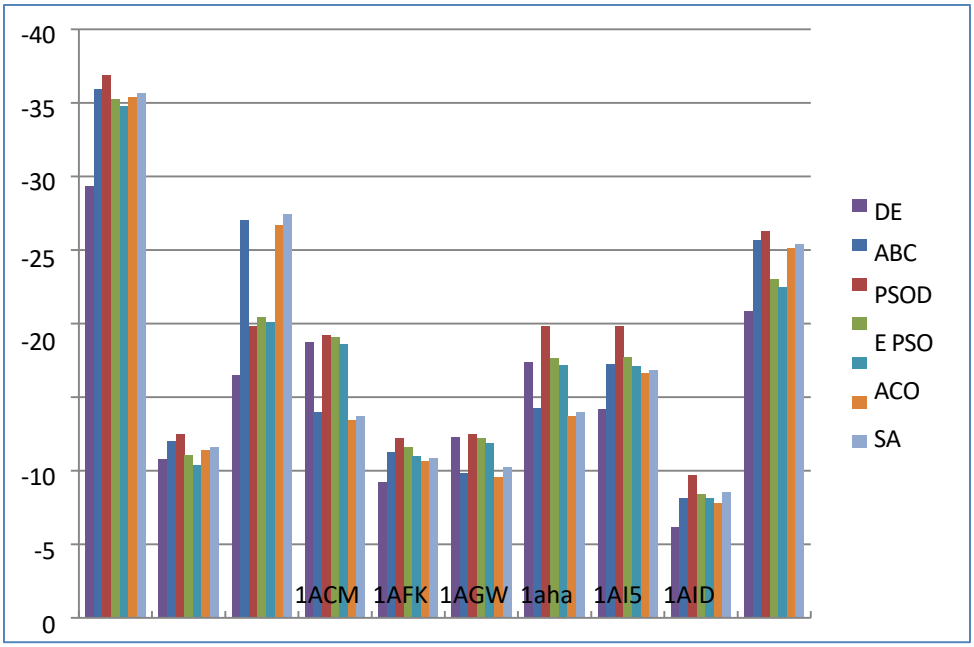


Figure 10 Protein ligand complex dataset with comparison of algorithms from 1ACM to 1B80

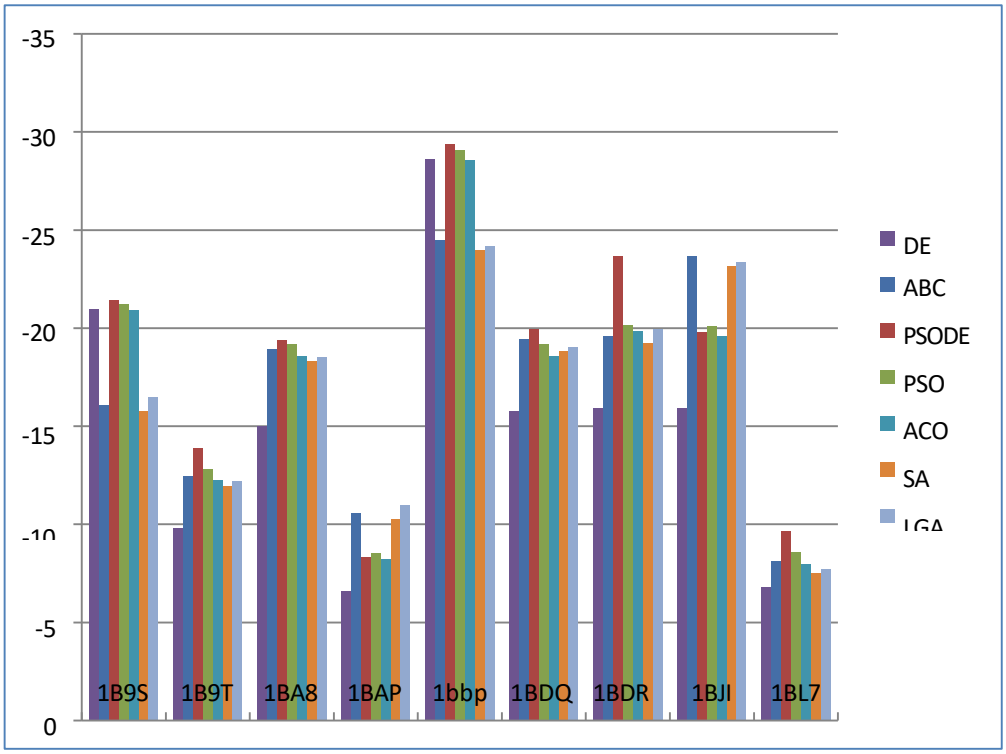
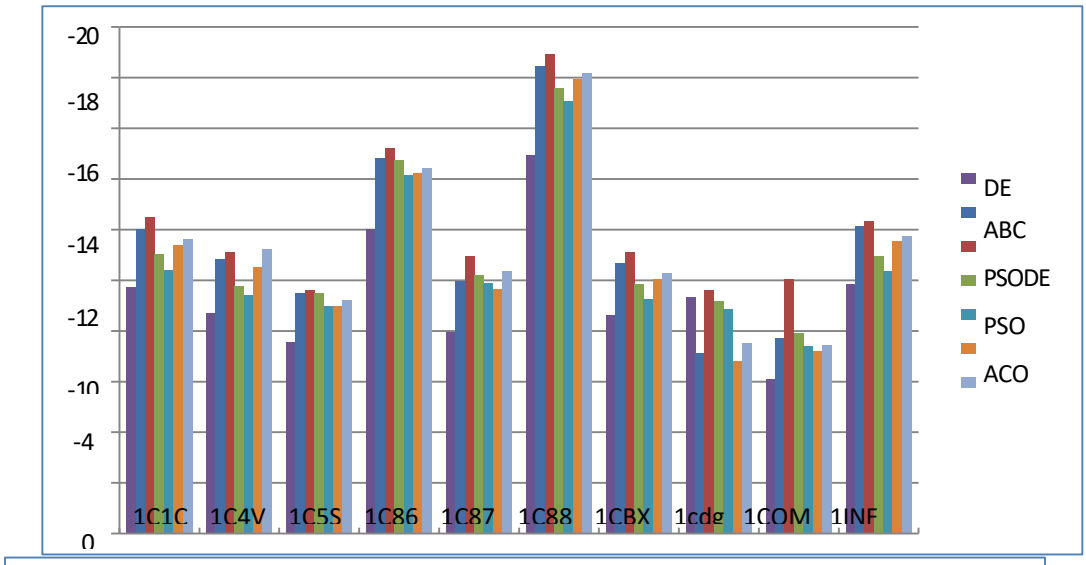


Figure 11 Protein ligand complex dataset with comparison of algorithms from 189S to 1BL7



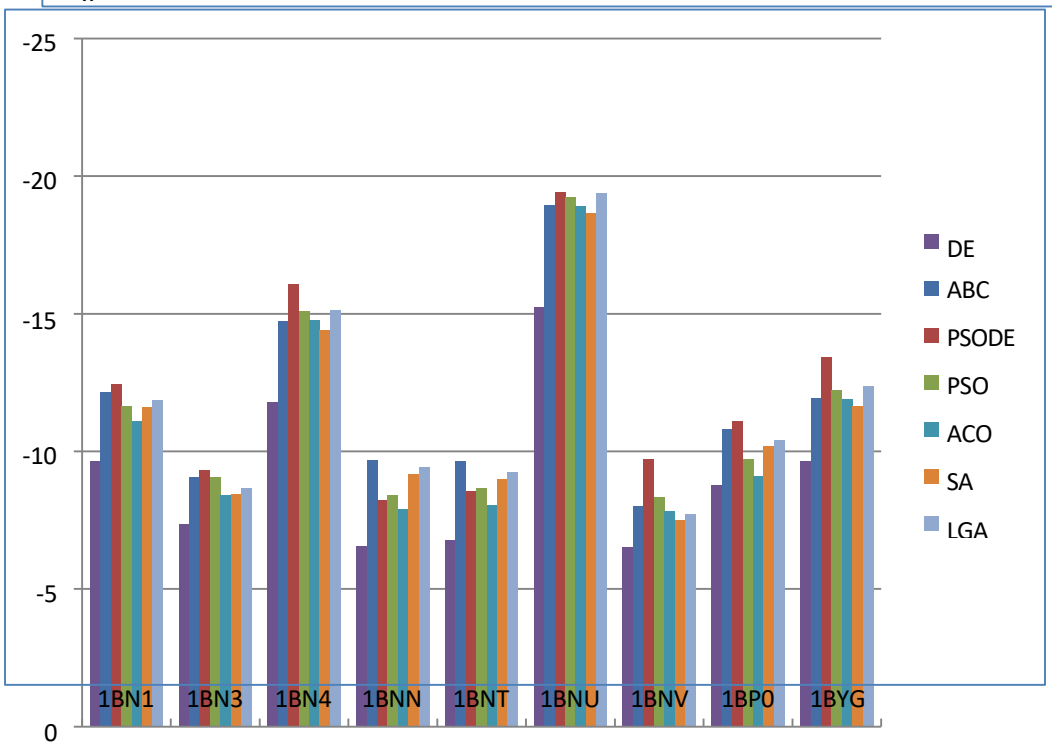


Figure 12 Protein ligand complex dataset with comparison of algorithms from 1BN1 to 1BYG

Figure 13 Protein ligand complex dataset with comparison of algorithms from 1C1C to 11NF

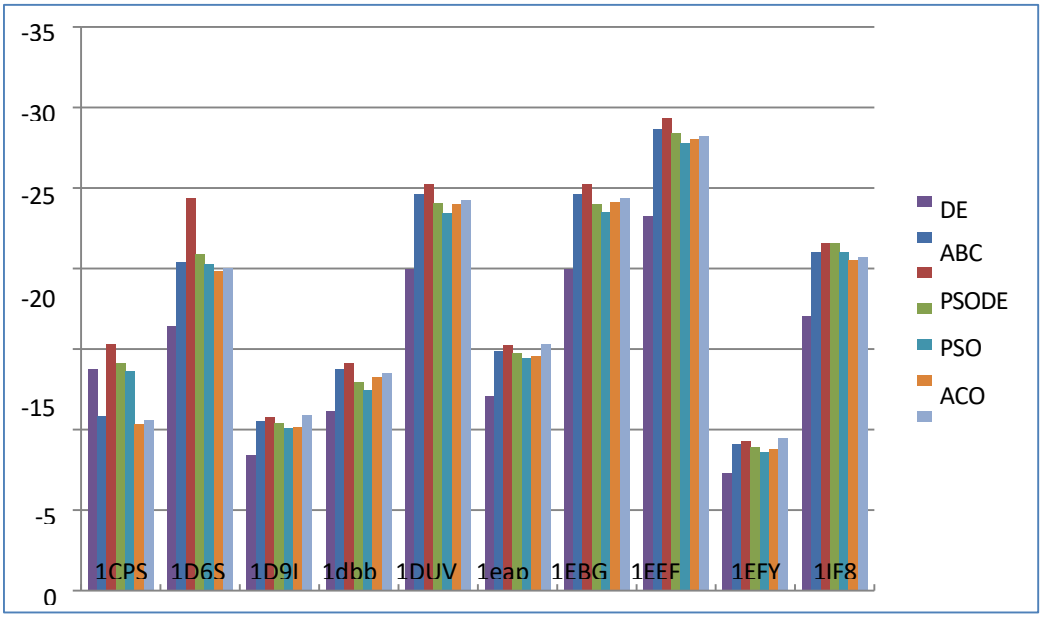


Figure 14 Protein ligand complex dataset with comparison of algorithms from 1CPS to 1IF8

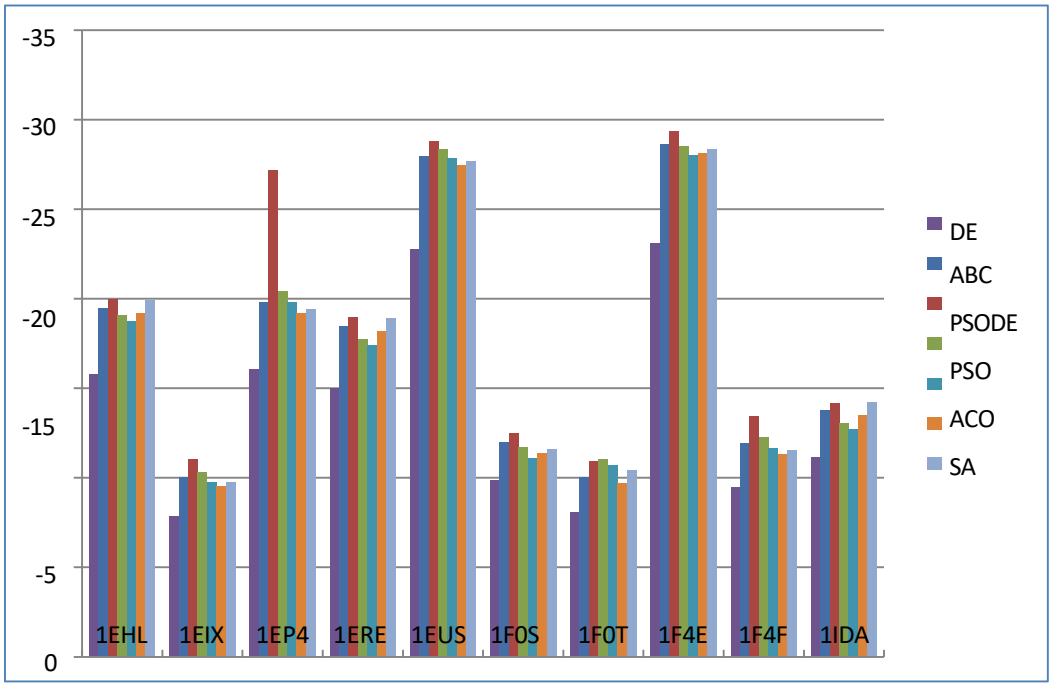


Figure 15 Protein ligand complex dataset with comparison of algorithms from 1EHL to 1IDA

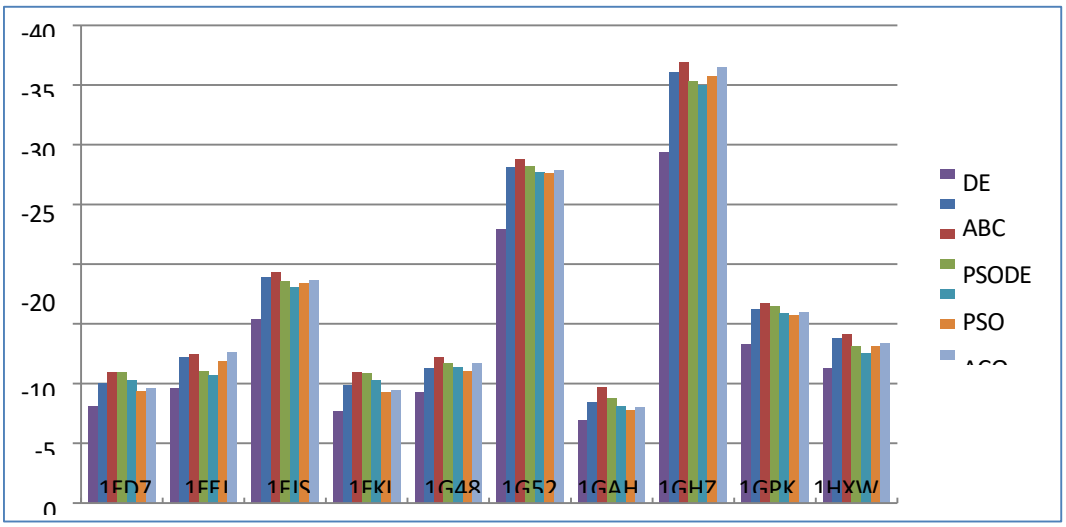


Figure 16 Protein ligand complex dataset with comparison of algorithms from 1FD7 to 1HXW

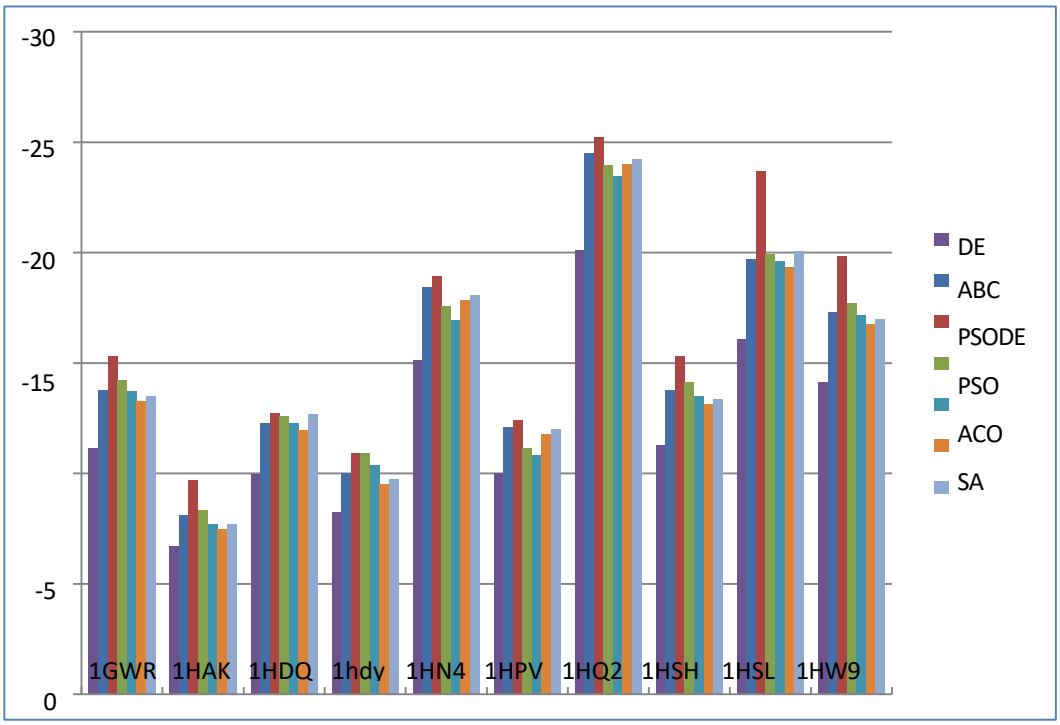


Figure 8 Comparison of DE, PSO, ABC and PSODE algorithm with docking energy.

Docking accuracy is evaluated depends on RMSD (root mean square deviation) of reference structure and obtained complex on docking .2 ˚A is set as a cut off value. Predicted binding mode is considered as perfect only when the RMSD value is below 1 ˚A.

Comparison of the least docking energy of DE, ABC, PSO, and PSODE

Protein Id	DE	ABC	PSO	PSODE
121P	-17.180	-21.647	-21.131	-21.550
182L	-15.252	-12.750	-15.624	-15.794
183L	-16.200	-27.136	-19.926	-20.412
185L	-7.110	-8.745	-8.959	-9.784
186L	-6.700	-8.241	-8.442	-9.805
187L	-9.970	-12.263	-11.798	-12.562
1A28	-13.280	-16.733	-16.334	-16.557
1A4G	-5.200	-6.396	-6.552	-7.950
1A4Q	-6.950	-8.549	-8.757	-9.763
1A52	-11.350	-13.961	-14.301	-15.370
1A5S	-17.180	-21.131	-21.550	-21.647
1A8I	-13.280	-16.733	-16.334	-16.557
1A9M	-12.180	-14.981	-14.893	-15.347
1aaq	-20.418	-15.550	-20.916	-24.486
1abe	-10.025	-7.990	-10.269	-10.706
1ABE	-29.380	-36.137	-35.404	-37.019
1ABF	-15.240	-18.745	-19.202	-19.292
1ACJ	-7.480	-9.425	-9.063	-9.200
1acm	-11.414	-9.950	-11.693	-12.307
1ACM	-29.380	-36.137	-35.404	-37.019
1AFK	-10.000	-12.300	-11.141	-12.600

Comparison of the lowest docking energy of DE, ABC, PSO, and PSODE

Protein Id	DE	ABC	PSO	PSODE

1AGW	-16.200	-27.136	-19.926	-19.926
1aha	-18.745	-13.950	-19.292	-18.745
1AI5	-9.280	-11.414	-11.693	-12.307
1AID	-12.300	-9.950	-12.600	-12.300
1AJV	-17.392	-14.250	-19.928	-17.392
1AZM	-14.140	-17.392	-17.816	-19.928
1B8O	-6.700	-8.241	-8.442	-9.805
1B9S	-21.131	-16.070	-21.647	-21.131
1B9T	-10.150	-12.485	-12.789	-13.907
1BA8	-15.500	-19.065	-19.345	-19.530
1BAP	-6.790	-10.579	-8.555	-20.412
1bbp	-28.757	-24.480	-29.459	-28.757
1BDQ	-15.930	-19.594	-19.197	-20.072
1BDR	-16.080	-19.778	-20.261	-23.786
1BJI	-16.080	-23.786	-19.778	-20.261
1BL7	-6.950	-8.549	-8.757	-9.763
1BN1	-9.970	-12.263	-11.798	-12.562
1BN3	-7.480	-9.200	-9.063	-9.425
1BN4	-11.970	-14.723	-15.082	-16.059
1BNN	-6.700	-9.805	-8.241	-8.442
1BNT	-6.950	-9.763	-8.549	-8.757
1BNU	-15.500	-19.065	-19.345	-19.530
1BNV	-6.700	-8.241	-8.442	-9.805
1BP0	-8.900	-10.947	-9.879	-11.214
1BYG	-9.800	-12.054	-12.348	-13.536
1C1C	-10.000	-12.300	-11.141	-12.600

Comparison of the lowest docking energy of DE, ABC, PSO, and PSODE

Protein Id	DE	ABC	PSO	PSODE
1C4V	-8.900	-10.947	-9.879	-11.214
1C5S	-7.720	-9.496	-9.487	-9.727
1COM	-6.280	-7.724	-7.913	-10.059
1CPS	-13.899	-10.850	-14.238	-15.423
1D6S	-16.600	-20.418	-20.916	-24.486
1D9I	-8.530	-10.492	-10.420	-10.748
1dbb	-11.290	-13.887	-13.123	-14.225
1DUV	-20.120	-24.748	-24.136	-25.351
1eap	-12.180	-14.981	-14.893	-15.347
1EBG	-20.120	-24.748	-24.136	-25.351
1EEF	-23.380	-28.757	-28.514	-29.459
1EFY	-7.480	-9.200	-9.063	-9.425
1EHL	-15.930	-19.594	-19.197	-20.072
1EIX	-8.160	-10.037	-10.282	-10.992
1EP4	-16.200	-19.926	-20.412	-27.136
1ERE	-15.110	-18.585	-17.723	-19.039
1EUS	-22.930	-28.204	-28.355	-28.892
1F0S	-9.970	-12.263	-11.798	-12.562
1F0T	-8.240	-10.135	-11.024	-11.024
1F4E	-23.380	-28.757	-28.514	-29.459
1F4F	-9.800	-12.054	-12.348	-13.536
1FD7	-8.240	-10.135	-11.024	-11.024
1FEJ	-10.000	-12.300	-11.141	-12.600
1FJS	-15.360	-18.893	-18.561	-19.354
1FKI	-8.240	-10.135	-11.024	-11.024
1G48	-9.280	-11.414	-11.693	-12.307

1G52	-22.930	-28.204	-28.355	-28.892
1GAH	-6.950	-8.549	-8.757	-9.763
1GHZ	-29.380	-36.137	-35.404	-37.019

Comparison of the lowest docking energy of DE, ABC, PSO, and PSODE

Protein Id	DE	ABC	PSO	PSODE
1GPK	-13.280	-16.334	-16.557	-16.733
1GWR	-11.300	-13.899	-14.238	-15.423
1HAK	-6.700	-8.241	-8.442	-9.805
1HDQ	-9.980	-12.275	-12.575	-12.752
1hdy	-8.240	-10.135	-11.024	-11.024
1HN4	-15.110	-18.585	-17.723	-19.039
1HPV	-10.000	-12.300	-11.141	-12.600
1HQ2	-20.120	-24.748	-24.136	-25.351
1HSH	-11.300	-13.899	-14.238	-15.423
1HSL	-16.080	-19.778	-20.261	-23.786
1HW9	-14.140	-17.392	-17.816	-19.928
1HXW	-11.290	-13.887	-13.123	-14.225
1IDA	-11.290	-13.887	-13.123	-14.225
1IE9	-20.850	-25.646	-23.002	-26.271
1IEP	-15.110	-18.585	-17.723	-19.039
1IF8	-17.180	-21.131	-21.550	-21.647
1INF	-10.000	-12.300	-11.141	-12.600
1C86	-12.180	-14.981	-14.893	-15.347
1C87	-8.100	-9.963	-10.206	-10.971
1C88	-15.110	-18.585	-17.723	-19.039
1CBX	-8.900	-10.947	-9.879	-11.214
1cdg	-9.496	-7.130	-9.272	-9.727

Table 1. Comparison Of The Least Docking Energy of DE, PSO, ABC And Psode Algorithm.

Figure 8 shows the rate RMSD(root mean square deviation) values for all docking programs is compared to 100 cases. This confirms PSODE algorithm is the efficient algorithm for docking search spaces. PSO and LGA predict the ligand position which is located long from the actual ligand binding site. Contrariwise, PSO and ABC algorithms have high RMSD value of 2.04 ˚A and 2.31 ˚A, which means that it calculates least energy for most of the cases but the deviation of predicted and pre tested binding pose is highly differentiate .For the 100 complexes, PSODE predict the binding position with low rate of RMSD. This reveals that proposed PSODE algorithm is appropriate for high dimensional docking applications.

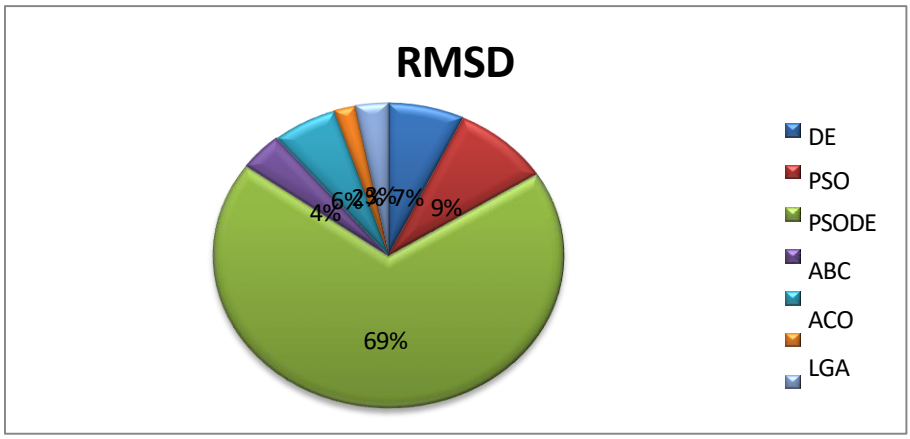


Figure 9 Comparison of RMSD of DE, PSO, ABC and PSODE algorithm.

In drug designing and discovery studies, Protein-ligand docking is a core one. Multidimensional docking is a significant problem in protein-ligand docking because of inflated conformational search space. In this research, evolution-based particle swarm algorithm is proposed to solve the protein-ligand docking problem. Compared with PSO, ABC, DE, and PSODE, hybrid PSODE show docking accuracy with optimal results. ABC predicts a better energy value but rmsd accuracy is not effectual. To summarize, PSODE appears well matched for exact evaluation for molecular docking.

4. Conclusion

This paper represents the protein ligand docking methods, framework for proposed hybrid algorithm and classifies the docking results obtained from the previous algorithm used. A Framework is presented for future development work of this research. Based on these results, it is determined that the differential evolution-based particle swarm optimization (PSODE) algorithm provides better results than the existing protein ligand docking algorithm. Hence this research provides a substitute method for a molecular docking. The future enhancement of this research may concentrate on increasing the computing speed, and evaluating energy based on multi-objective function.

REFERENCES

1. Sousa S.F., Fernandes P.A., Ramos M.J. (2006) Protein-ligand docking: current status and future challenges. Proteins; 65:15–26.
2. Dr.P. Lakshmi, Dr.D. Ramyachitra, “An improved Genetic with Particle Swarm Optimization Algorithm Based on Ensemble Classification to Predict Protein – Protein Interaction “, Wireless Personal Communications, Springer, 18 April 2020, volume 113, Issue 4, 1851-1870.
3. Dr.P. Lakshmi, Dr.P. Manikandan, Dr.D. RamyaChitra, “Motif Discovery of Protein-Protein Interaction

Networks by using Penalized Matrix Decomposition with Bagging of Machine Learning Algorithms “, IEEE Access, Volume 13, 2025

4. Stahura F.L., Jaborah J. (2004) Virtual screening methods that complement HTS. Comb Chem High Throughput Screen;7:259– 269.
5. Klebe G. (2006) Virtual ligand screening: strategies, perspectives and limitations. Drug Discov Today; 11:580–594.
6. Gnther R., Elsner C., Schmidt S., Hofmann H.J., Bordusa F. (2004) On the rational design of substrate mimetics: the function of docking approaches for the prediction of protease specificities. Org BiomolChem; 2:1442–1446.
7. Lyne P.D. (2002) Structure-based virtual screening: an overview. Drug Discov Today; 7:1047–1055.
8. Kuntz I.D., Blaney J.M., Oatley S.J., Langridge R., Ferrin T.E. (1982) A geometric approach to macromolecule-ligand interactions. J Mol Biol; 161:269–288.
9. Jiawi H. and Michelin K.,” Data Mining: Concepts and Techniques”, Academic Press, 2001
10. Wang R., Wang S. (2001) How does consensus scoring work for virtual library screening? An idealized computer experiments. J Chem Inf Model; 41:1422–1426.
11. Perola E., Walters W., Charifson P. (2004) A detailed comparison of current docking and scoring methods on systems of pharmaceutical relevance. Proteins; 56:235–249.
12. Kirkpatrick S., Gelatt C.D. Jr, Vecchi M.P. (1983) Optimization by simulated annealing. Science; 220:671– 680.
13. Holland J.H. (1975) Adaptation in Natural and Artificial Systems. Ann Arbor, MI, USA: University of Michigan Press.
14. Morris G.M., Goodsell D.S., Halliday R.S., Huey R., Hart W.E., Belew R.K. et al. (1998) Automated docking using a Lamarckian Genetic Algorithm and an empirical binding free energy function. J ComputChem; 19:1639–1662.
15. Kennedy J., Eberhart R.C. (1995) Particle swarm optimization. Proc IEEE Int Conf Neural Netw (ICN95); 4:1942–1948.
16. Chen H.-M., Liu B.-F., Huang H.-L., Hwang S.-F., Ho S.-Y. (2007) SODOCK: swarm optimization for highly flexible protein-ligand docking. J ComputChem; 28:612–623.
17. Meissner M., Schmuker M., Schneider G. (2006) Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training. BMC Bioinformatics; 7:125.
18. Clerc M. (1999) The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. Proc Cong Evol.Comput (CEC99);3:1951–1957.
19. Eberhart R.C., Shi Y. (2000) Comparing inertia weights and constriction factors in particle swarm optimization. Proc IEEE Congress EvolComput (GEC00); 1:84–88.
20. Liu Y., Qin Z., He X. (2004) Supervisor-student model in particle swarm optimization. Proc IEEE Cong EvolComput (CEC04); 4:542– 547.
21. Riget J., Vesterstroem J.S. (2002) A diversity-guided particle swarm optimizer – the ARPSO. Technical Report 2002-02. Aarhus: Department of Computer Science, University of Aarhus p. 13.
22. Hart W.E. (1994) Global Optimization with Local Search. PhD Thesis. San Diego: University of

California; p. 151.

23. Solis F.J., Wets R.J.B. (1981) Minimization by random search techniques. Math OperRes; 6:19–30.
24. Stouten P.F.W., Frçmmel C., Nakamura H., Sander C. (1993) An effective solvation term based on atomic occupancies for use in protein simulations. Mol Simul; 10:97–120.
25. Lam P.Y., Jadhav P.K., Eyermann C.J., Hodge C.N., Ru Y., Bacheler L.T. et al. (1994) Rational design of potent, bioavailable, nonpeptide cyclic ureas as HIV protease inhibitors. Science;263: 380–384.
26. Sanner M.F. (1999) Python: a programming language for software integration and development. J Mol Graph Model; 17:57–61.
27. Berman H.M., Westbrook J., Feng Z., Gilliland G., Bhat T.N., Weissig H. et al. (2000) The Protein Data Bank. Nucleic Acids Res; 28:235–242.
28. Gasteiger J., Marsili M. (1980) Iterative partial equalization of orbital electronegativity – a rapid access to atomic charges. Tetrahedron; 36:3219–3228.