International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Integration of Diversity Enhancement of Particle Swarm Optimization and Neighbourhood Search with k radius to predict Software Cost Estimation

V Venkataiah[1], M Nagaratna[2], Ramakanta Mohanty[3]

Submitted: 06/06/2022 **Accepted**: 10/09/2022

Abstract

Prediction of software development cost is a crucial activity in software engineering community at early stages of software development of life cycle. It helps to project manager to do better project management i.e. effective planning,organizing and monitoring. Generally, inaccurate estimation cost due to lack of data and inherent relationship between attributes. For accurate software cost estimation, the amount of techniques has been proposed, one of them is Particle Swarm Optimization (PSO) has been exposed an impressive performance. However, it is struck at local minima due to diversity loss quickly. In order to improve its searching ability and convergence rate, this paper proposes a new hybrid approach iscalled DPSONS-K. It consists a diversity enhanced method and neighborhood search methods with k radius. Where k is tuning parameter used to achieve stability between searching and convergence abilities.Seven benchmark datasets are used to investigate outcome of proposed approach.Comparative study shows that DPSONS-k approach achieved better results than other ones.

Keywords: Particle Swarm Optimization (PSO), Diversity Enhanced Particle Swarm Optimization (DPSO), Neighborhood Search (NS), Root Mean Square Error (RMSE).

1. Introducton

Food is an essential for living things; nowadays the software is an essential to the society. But is it developed as per client requirements are best quality, low cost, and within the budget; A report from the Standish Group's Chaos [1], 66% of the projects are completed within the time bound, appropriate cost, and better quality, and 34 % projects either overestimation or underestimation that leads to financial loss. While developing software products faces a number of problems at the early stages of the software life cycle, one of them is inefficient resource planning which leads to failure of the project. It is a process of predicting efficient procedures, resources to complete the software development project is software cost estimation. Basically, it deals with doing rough estimation at the early stage of the project life cycle which including requirements, planning, design, implementation, and maintenance. But at this stage required data is not available and every project has own unique property which makes difficulty to calculate required estimation to complete the project. Authors in [2] [3] suggested that upcoming models will have acclimate good number of projects because of incomplete project data. The real fact is that software project data sets are typically small, and their original relations are unpredictable or missing. Due to this prediction of accurate effort estimation is challenging task.

Number of methods that have been proposed to predict accurate estimation can be classified into three types are 1) algorithm models are based on mathematical formula best examples are Constrictive Cost Model (COCOMO) [4], Putnam' and Function Point Analysis (FPA) [5] and are suitable for less complex software development product. 2) Non-algorithms models are based on experience, expert knowledge, and similar kind of projects for instance [6] Expert Judgement, Top-

[1] Associate Professor, Dept. of CSE, CMR College of Engineering & Technology, Hyderabad, India.
[2] Professor, Dept. of CSE, JNTUH College of Engineering, Hyderabad, India.
[3]Professor, Dept. of CSE, Swami Vivekananda Institute of Technology, Secunderabad, India.

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Down, Bottom-Up, Price to win, and Analogy. 3) Learning-oriented models are learning from the set of projects and examining, making decisions corresponding to new projects. Artificial Neural Networks (ANN) [7, 8], Fuzzy Logic (FL) [9, 10, 11], Genetic Algorithms (GA)[12,13], Support Vector Regression, (SVR)[14], Basian Networks (BN) [15] and Regression Tree (RT) [16]. None of these can fit all kinds of projects. Nowadays, Software engineering problems considered as searching problems were solved using many variants of swarm intelligence based algorithms that have been proposed for three decades, e.g. Particle Swarm Optimization (PSO) [17], Ant Colony Optimization (ACO) [18], Chemical Reaction Optimization (CRO) [21],Cat Swarm Optimization (CSO) [19], Artificial Bee Colony Optimization (ABCO) [20], and it has come more popular in swarm intelligence community due to easy implementation, yeteffectiveness.It is notified that PSO outcome is extremelyassociated to the diversity of the particles, particularly when attemptsare finished avoiding the early convergence at local minima. Hence, PSO provides better search and convergence abilities, and maintainshigher diversity in the swarm.

The aim of this paper is providing methodologies for enhancing diversity and local and global search with k parameter tuning of particle swarm optimization to predict accurate cost estimation of software project at early stages of the software development which gives benefits to customer and as well as project manager to better understand. This paper can be organized as follows. Section 2 presented background works. Section 3 describes our new approach.Section 4 showsinvestigational results and discussions. Finally, concluded the proposed work, and direction tofuture work in Section 5.

2. Background Work

Puspaningrum et al. [22] proposed a novel model of cuckoo optimization and harmony search algorithm. Langsari et al. [23]andUllah et al. [27] PSO andFlower Pollination Algorithm (FPA)modelsare used to tuning the COCOMO II coefficients. Ahadi et al. [24] PSO and DE algorithms are employed to provide more comprehensive and efficient estimate. Padmaja et al. [25] offered a model which uses Grey Relational Analysis (GRA) to predict accurate effort estimation,and later experimental result shows best minimum error value compared to traditional techniques.Nassif et al. [26] presented three fuzzy logic models are Mamdani, Sugeno with constant output, and Sugeno with linear output, were used to design, evaluate, and compare estimated efforts. Khazaiepoor et al. [28] proposed technique is presented as three steps. In first step, integration of neural networks and genetic algorithms are used to select relevant features. Impact factors are computed by multiple linear regressions in second step. In third step analyze optimized feature weightsby Imperialist Competitive Algorithm (ICA). Shahpar et al. [29] PSO-SA a novel hybrid technique used to find out optimized feature weights widely utilized in similarity functions are Euclidean, Manhattan, and Minkowski for Analogy based estimation. Venkataiah et al. [30] proposed a novel hybrid model iscombination of PSO and K-means is used to predict optimized software cost estimation.

The basic PSOwas invented by Eberhart and Kennedy in 1995[31], it is a biological inspired evolutionarycomputation [32] and solution based stochastic search technique [33] start with an initial solutionsare generated randomly [34], apply swarm theory, update position of the particles, and find out best solution. It is required less memory, computationally inexpensive, and easier to implement [33]. Each particle in a swarm can be represented by its position and velocity in D-dimensional space for the search problem.Each particle is attracted by its personal best (pb) and the global best (gb) in during a search process as followed [34].

$$v_{ij}(t+1) = w.v_{ij}(t) + c_1.r_1 + \left(pb_{ij}(t) - x_{ij}(t)\right) + c_2.r_2 + \left(gb_{ij}(t) - x_{ij}(t)\right) \quad (1)$$

$$x_{ij}(t+1) = v_{ij}(t+1) + x_{ij}(t) \quad (2)$$

where i= 1 to N is an index of the particle , $X_i = (x_{i1,}x_{i2,} \ldots\ldots, x_{iD})$ is the ith position,$V_i = (v_{i1,}v_{i2,} \ldots\ldots, v_{iD})$ is the ith particle velocity, $pb_i = (pb_{i1,}pb_{i2,}.., pb_{iD})$is the best previous position give up the best fitness value of the ith particle; and $gb_i = (gb_{i1,}gb_{i2,}.., gb_{iD})$ is the global best particlesfrom the swarm. The parameter w, known as inertia weight factor, which is used to control the direction

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

of velocity [34]. The most popular optimizer is PSO has widely utilized innumerousrealistic applications since it was invented. PSO has many variants have been proposed from past thirty years. A summarized variant of PSO is presented has follows. Yuhui Shi and Russell Eberhart [35] inertia weight w, andconstriction factor Clerc [36] and [37, 38] were used to solve different applications. Bergh and Engelbrecht [39] presented a formal proof which describes each particle converges to a stable point. Kennedy [40] proposed fourdissimilar types of neighborhood of topologies and his experimental result shows that complex problems may give better performance with small Neighborhood of PSO, while smaller problems may give better output with large Neighborhood of PSO. Shi et al. [41] promotesthat particle in PSOencouraging exploration at early stage, in the next stages of the search, fine tuning is becomingmandatory because of get best solution which implies that exploitation search is required. Suganthan[42] first presented a dynamic neighborhoodsearch operator which is gradually elaborate neighborhood size step by step until it completes all the solutions in a swarm. Hu et al. [43] dynamic neighborhood PSO is updated neighbours of each particle by calculate distance between the present and others particle, find the nearest n particles which are nearby current particle. Mendes et al. [44] introduced a Fully Informed PSO algorithm (FIPSO), according to FIPSO, Mohais et al. [45] dynamic neighborhood is presentedby re-organizing neighborhood in terms of a diversity-stabilitymeasure. Berghet al. [46] highlighted is that the original PSO performance enhanced by cooperative behaviour is known as CPSO-H. Lang et al. [47] proposed Comprehensive PSO (CPSO);Chen et al. [48] projected a novel PSO with Dynamic Linkage Discovery which is used to solve the linkage problem in parameters of any optimization technique. Hsieh et al. [49] presented an Efficient Population Utilization Strategy for PSO (EPSO). Cervantes et al. [50] introduced a novel PSO is called Adaptive Michigan PSO (AMPSO) employed to dimensional reduction of search space and provided more flexibility than the former one. Zhan et al. [51] proposed an Adaptive PSO (APSO), Wang et al. [52] Self-adaptive learning based PSO strategy used to enhance CLPSO performance. Wang et al. [53] present a novel an improved comprehensive particle swarm optimization by using a Generalised Opposition- Based Learning (GOBL).

3. Hybrid Approach DPSONS-K

In this section, innovative variant of PSO is called Integration of PSO diversity enhancement and neighborhood search with K radius (DPSONS-K) is proposed work. The DPSONS-K employs two approaches including PSO diversity enhancement and neighborhood search with k radius.

3.1 PSO Diversity Enhancement

Standard Particle Swarm Optimization is used to test multi-model problems, which are be likely suffering from the local minima and premature convergence due to diversity decrease in a search problem space that guides to total implosion and fitness stagnation of swarm. In order to overcome this problem of PSO, S. Das et al. [54] proposedtwo phasesareAttractive and Repulsive in PSO is called (ARPSO). In phase one all the particles are attracting each other because of information sharing is good between particles, and hence the diversity decreases. In phase two individual particles are no longer attracted that means repelled by overall best position and best local position. If the diversity is less than lower bound $d_l$then it switches to R (repulse) phase,otherwise it switches back toA (attraction) phase. In a whole process search behaviour changes from lower to upper and vice versa by ARPSO. The search behaviour does not alter by the ARPSO when diversity remains between d_l and d_h. Pant et al. [55] proposed a trail phase is known as middle phase between A and R. In middle phase, it is neither completesA nor complete R. Each particle is attracted by its own best position and is repelled by the best overall best particle.

$$\text{DIVERSITY}=\frac{1}{N}\sum_{i=1}^{N}\sqrt{\sum_{j=1}^{D}\left(x_{ij}(t)-\underline{x_{ij}}(t)\right)^2} \quad (3)$$

$$\underline{x_{ij}}(t)=\frac{\sum_{i=1}^{N} x_{ij}(t)}{N} \quad (4)$$

To avoid calculation of the diversity and control the swarm divergence, Wang et al. [53] propose a diversity improved method. PSO position updated rule for every particle $AP_i(t)$, an ovel particle $AP_i(t+1)$ is generating.

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

By recombining $AP_i(t)$, and $AP_i(t+1)$ a trail particle $BP_i(t+1) = (BX_i(t+1), BV_i(t+1))$ is created as follows:

$$BX_{ij}(t+1) = \{X_{ij}(t+1), \quad if\, r_j(0,1) < P_{r1}\, X_{ij}(t), \quad otherwise \qquad (5)$$

$$BV_{ij}(t+1)) = V_{ij}(t+1) \qquad (6)$$

From the equation (5) it is observed that $BX_{ij}(t+1)$ is not effective position vector when Pr1=0% and 90%, due to this there is chance to lose the diversity between particles of the swarm, and random number behaviour it not positive one. Apply shuffling to position vector index after generating in iteration. Hence, improve the diversity between particles of the swarm.

$$RP_i(t+1) = shuffle(BP_i(t+1)) \qquad (7)$$

Where i=1 to N, j=1 to D $r_j$is a random value over (0, 1), and P_{r1} *is a* predefinedprobability value. Every element in $BP_i(t+1)$ position vector obtains from the $AX_i(t+1)$and $AX_i(t)$with probabilities of $P_{r1}$and 1-P_{r1} respectively. This is same to crossover technique of Differential Evaluation (DE) by Stornet al. [56]. A clear graphical representation of how to generate $BP_i(t+1)$ is shown in Fig.1, and Fig.3 $RX_i(t+1)$ respectively.

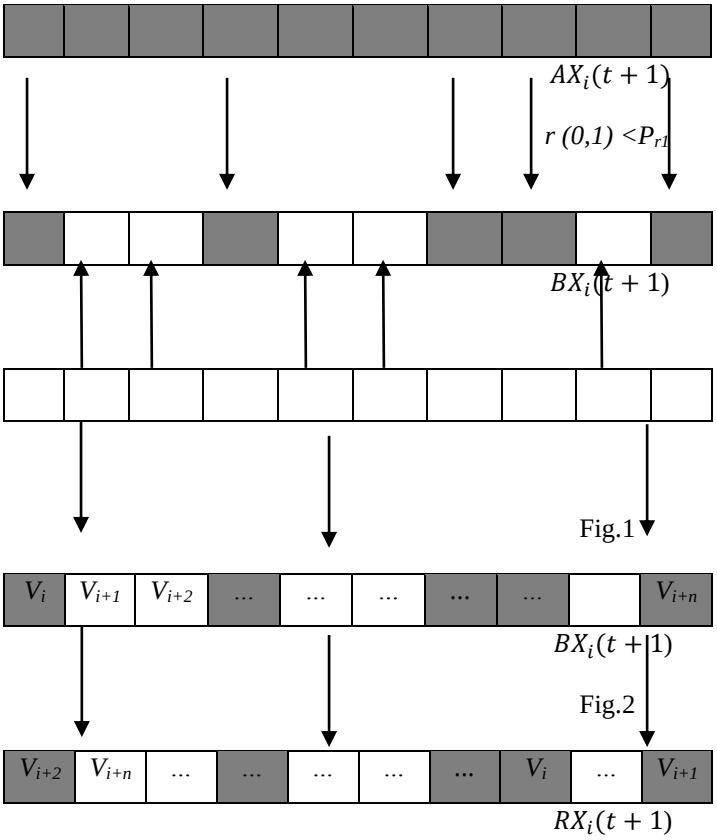

Figure.3

After recombination and shuffling, a greedy selection procedure is used as follows:

$$AP_i(t+1) = \{RP_i(t+1), \quad if\ f\left(RP_i(t+1)\right) \le f\left(AP_i(t+1)\right), \ AP_i(t+1), \quad otherwise \qquad (8)$$

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Wherethe fitness evaluation function is f(.),this paperconsidered minimization problemsonly. If and only if,a new generated particle $RP_i(t+1)$ isbetterthan $AP_i(t+1)$, then $AP_i(t+1)$ replace with $RP_i(t+1)$, otherwise, $AP_i(t+1)$ is remains same as previous.In general particles are frequently to move same direction in the search process of PSO. it means with increasing iterationssame kind of particles become.The trail $RP_i(t+1)$is selected into next generation due to dissimilarities between $RP_i(t+1)$ and $AP_i(t+1)$. Ifdissimilarities are more in the swarm means that higher diversity. Hence, to control diversity in the swarm,we need a control parameter is known as P_{r1}. If P_{r1} value is small, then higher diversity exists between presentand next generation and if P_{r1}value is larger diversity will be decreases.

3.2 Neighborhood Search with K Radius

Nowadays, all the complex problems like multimodal are solved using population-based searching algorithms which are suffering from the problem of premature convergence. Occasionally, the suboptimal solutions are close to bestsolution and the neighborhood of trapped individual solution may contain the bestsolution. In this scenario, searching of neighborhood particlesisuseful to determine the best solutions. Based on this concept, some extraordinaryneighborhood searching approaches are applied to evolutionaryalgorithms. [54, 57, 44, and 58].

Basically, PSO performancestrongly depends on the parameters and they are directly or indirectly showing effect on their results. In order to avoid the dependency problem by employing interconnection topologies which are proposed by Kennedy in 1995; According to information sharing between particles, and these topologies can be classified into two groups are called global and local neighborhood topology to trade of the exploration and exploitation of DE[44].

Suppose there are P_m particles in a swarm where m=1, 2, 3... N and it is also size of thepopulation. All the particle is arranged as a ring topology based on their indices, such that last particle (P_n) and second particle (P_2) are the two neighbours of first particle(P_1). There are 16 particles in the swarm as represents ring topology shown in Fig.4.Anyhow, there are different kinds of neighborhood topologies wheel, star, pyramid, and ring for PSO.The ring topology is quite a simple and easy to implement. For every particle isP_m, itsK-neighborhood radius consisting of P_{m-k}... Pm, .,P_{m+k} where K is an integer$0 \leqslant K \leqslant \frac{N-1}{2}$. All empirical studies consider K=2, but in this article, K value is variable, and it is also proposed work.

3.2.1Local Neighborhood Search Approach (LNSA)
For every particle, its neighborhoodgives better solutions.LNSA is used to enhance ability of the convergence which has been proposed byWang et al. [59]. When searching the near particle of Pm, a trail particle defined as L_j= (LX_j, LV_j) is obtained as defined.

$$LX_j = r_{11}\,.X_j + r_{12}\,.\,pb_j + r_{13}\,.(X_f - X_g) \quad (9)$$

$$LV_j = V_j \quad (10)$$

where X_j is jth particle position vector representation, pb_j is the last iteration best value of APj, X_f and X_g are random vector positions of particles in k-neighborhood radius of P_j, f, g ϵ [j-k, j+k] Λ f≠ g≠j, r_{11},r_{12},and r_{13} arerandom numbers within range of (0,1) and $r_{11}+r_{12}+r_{13}=1$. The random parameters r_{11}, r_{12}, andr_{13} aresame for all l=1, 2...D and they are generated in each new generation. More explanation of LNSA is depicted in Fig. 5a.The pb_j is last iteration best particle of X_j and it is not appeared on the circle. To maintain the same velocity for flying direction of AP_j and the trail particle LNSA $_i$ is more attractive when a local minimum is near to the global optimum. Particles are easily struck in to local optimalwith large jumps.

3.2.2Global Neighborhood Search Approach (GNSA)
This mechanism iscentre of attention on the global neighbour area of each particle to improve the ability of search. During search the neighbour area of a particle Pm, another trail particle Gj= (GXj, GVj) can be generated as follows.

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

$$GX_j = r_{14} \cdot X_j + r_{15} \cdot gb_j + r_{16} \cdot (X_p - X_q) \quad (11)$$

$$GV_j = V_j \quad (12)$$

where gb_j is the overall best particle, X_p and X_q are arbitrary position vector are selected for the complete swarm, p, qϵ[1, N] Λ p≠q≠j, r_{14},r_{13},and r_{16} are three arbitraryparameters within range of (0,1) and r_{14}+r_{15}+r_{16}=1 which aresame for all l=1toD.These parameters aregenerated for each aniteration. The GNSA is used to solve practical problems.All the particles are placed in different regions. Hence, if the present particle is fall into local minima,then remaining particles in other regions may pull out of thetrapped.

For both LNSA and GNSA strategies, the newly generated trial particles are Li and Gi, consider the same velocity of their parent Pi.This aims to precede the same moving direction and jumping step size of their parent.

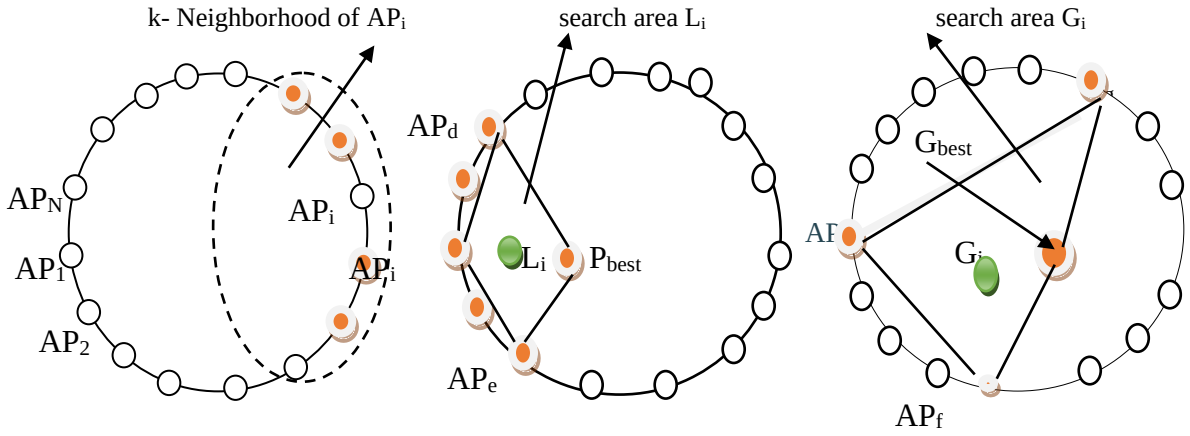

Fig 4. The ring topology K-Neighborhood (a) Local Neighborhood search (b) Global Neighborhood search

Figure. 5. Neighborhood search approaches

3.3. Proposed Approach

In this paper, we propose hybrid algorithmDPSONS-K which employs twostrategies including diversity procedure,neighborhood search operators with tuning Kparameter. WhereK is the radius of neighborhoodparticles. The previous studies were described firstindirectly swarmdiversity enhance, then afterconcentrate ontuning K parameter in a search process with neighbouring area.

In each iteration, AP_i trail particle is created by the recombination of previous and next generation of position vectors. Then after doing random shuffling the trail particle is to frame the RP_i. If at all RP_i is better than AP_i, and then replace AP_i with RPi; otherwise, AP_i is remains same. Once shufflingactivity is completed, and thenconducted aneighborhood search mechanism with P_{n1}probability on variousK parameter values in a tuning. Furthermore,If P_{n1}probability condition is satisfied, two Li and Gi trail particles are generated, then after, selectedwhich is the fittest particle amongthe Pi, Li and Gi as the new particle Pi.

Proposed DPSONS-K Algorithm1

1. Normalize data by min max normalizationrefer to Eq. (13);
2. Initialize each particle in the swarm randomly;
3. Calculate effort estimation according to theCOCOMOmodelrefer to Eq. (14);
4. Determine the fitness value of AP_iaccording toEq. (15);
5. pb and gb initialize;
6. while iteration<= max_iterarions:
7. for i in range (ps):
8. Calculatevelocity of each particle AP_i refer to Eq. (1);
9. Update position of each particle APi refer to Eq. (2);

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

10. Set boundaries of position APi;
11. Determine the fitness value of AP_i refer to Eq. (13);
12. Generate the new trail particle AP_i according Eqs. (5) & (6);
13. Generate the new trail particle RP_i according Eq. (7);
14. Calculate the fitness value of RP_i; according to Eq. (14);
15. Select a filter between AP_iandRP_ias a new AP_irefer Eq. (8);
16. Update pbt_i and gbt_i;
17end
18. for i in range (ps):
19. if rand (0,1) is less than or equal to P_{n1} then:
20. Generate new a trail $LCSA_i$ according to Eq. (9) & (10);
21. Generate new a trail $GCSA_i$according to Eq. (11)& (12);
22. Determine the fitness value of LCSAiandGCSAi refer to (15);
23. Select a filter between AP_i,$LCSA_i$and$GCSA_i$as a new AP_i;
24. end
25. Update pbt_i and gbt_i;
26. end
27. end

All the necessary steps are given in proposed DPSONS-K Algorithm1, where AP_i is the ith particle in a swarm, population size is N.Probability isP_{n1},iteration is number of iterations and max_iterarions is the total number of iterations. Outcome of this algorithm is in terms global best is root mean square error value indicates reduced gap between actual and estimated effort. Hence, we obtained accurate software cost estimation.

4. Results and Discussions

Conducted experimental study, in which 7 datasets are employed test outcome ofthe proposedmethod, details of seven datasets are described in Table 1. with statistical information. These datasets are taken from the literature [60, 61, 62, 63, 64, 65, and 66], then after do normalization using min-max normalization method as given below in Equation-(13), and Effort estimation calculated by using the basic COCOMO model as refer to Equation-(14) to apply proposed approach calculate accurate software cost estimation which is presented the following tables.

$$B' = \frac{(B-min_B)}{(max_B-min_B)}*(X-Y) +Y \quad (13)$$

Where $min_B$and max_B are the minimum and maximum values of attribute B, X=1, Y=0, andB' is mapping variable.

$$\text{Effort } (E_i) = a\,.\,(size_i)\,*^b \quad (14)$$

The computational configuration is listed as follows.

- System: Ubuntu 18.04 LTS ‘Xeniak Xerus’
- Memory: 3.7 GiB
- Processor: Intel Core i3-6100U CPU @ 2.30GHz
- Language: Python 2.6

Table 1. Data Sets Description

SL No.	Dataset Name	Features	Observations	Min Effort	Max Effort	Mean Effort	Standard Deviation	Unit
1	COCOMO81	16	63	5.9	11400	683.32	1821.58	PM
2	ALBRECHT	07	24	0.5	105.2	21.87	28.41	PM
3	CHINA	09	499	26	54620	3912.04	6480.85	PM
4	DESHARNAIS	11	81	546	23940	5045.30	4418.76	PM
5	ISBSG	105	4106	4	645694	5356.38	19789.68	PM
6	MAXWELL	25	62	583	6369	8223.21	10499.99	PM
7	MIYAZAKI94	9	46	5.6	1586	87.475	228.7585	MM

Table2. described the results obtained by standard alone DSPO at different levels of probabilities of p_{r1}on seven datasets.Root mean square error values

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

are hardly converge to the global optimum. Similarly, Results of root mean square error values are achieved by standard alone NSPSO with various levels of probabilities of P_{r1}on seven datasets are not impressed which are shown in Table 3.

The performance of DPSONS-K may be depending on the values of P_{r1} andP_{n1}. To choose best combination of these two control parameters,verified outcome of DPSONS-K mechanism under variant of P_{r1} andP_{n1}probabilities. All the experiments were conducted 20 times, androotmeansquareerror valuesare global optimum (gb) were recorded.

$$\text{RSME}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(E_i-\bar{E}_i)^2} \quad (15)$$

Where n is the number of data samples, $\underline{E_i}$ actual effort, $E_i$estimatedeffort is calculated according to equation-(13), a, b are parameters of COCOMO generated randomly with range of (-10, 10), size is represented by Kilo Lines of code. Average root mean square values are presented in Table 4.P_{r1}=0% means low diversity and P_{r1}=100% shows high diversity.

The proposed DPSONS-Kapproach makes use of twoapproaches: diversity improved mechanism,tuning K in a search with neighbouring area. To examine the effects twoapproaches, we studied the standard PSO, DPSO, and NSPSOwith tuningofKparameter. This approach helps to verify the efficiency of these strategies separately.The parameters P_{r1}=90%andP_{r1}=80%are used in DPSO, NSPSO, and DPSONS-KmechanismwhereK=1. For other parameters, we used the same parameters setting.

To pick the best groups of parameters setting at a statistical level, ranking of Friedman test conducted according to recommendations of [13,17], Table 5. shows rankingofDPSONS-Kapproachwith different parameter setting on differ radius of the K, where K=1, 2, 3, 4, and 5, and average ranking of Friedman test shown in Table6. From these P_{r1}=90%andP_{r1}=80%atK=1 achieves the lowest ranking. It demonstrates that P_{r1}=90%andP_{r1}=80%at K=1 are the relatively best choices for testing.

Table 7. describes average root mean square error values of eight algorithms. From the comparison of NSPSO with PSO, PSOW, PSOCL, and PSOWFIP. NSPSO achieved best results. When DPSO compared with PSO, PSOW, PSOCL, and PSOWFIP, it is hardly improved quality of the results. PSOU compared with the DPSO, surprising PSOU achieved best results than DPSO on majority of the data sets. By combing DPSO and NSPSO strategies, proposed mechanism obtains best performance. It is outperformance than other seven variants of PSO on majority of the datasets. Both NSPSO and DPSO are struckat local minima, while proposed approach achieves satisfactory results. The diversity enhanced mechanism is helps to increase diversity of the swarm, while it is slow down the convergence rate. The neighborhood search is based on the attraction of pb or gb. This helps to accelerate the convergence rate of PSO but runs the risk of losing swarm diversity. By integration of these two strategies, proposed approach achieves a balance between the exploration and exploitation abilities. Finally, Fig.6 describes results of DPSO with various Pr1 values. The Pr1 value at 60% most of the RMSE values achieved best results. Similarly, Fig.7 shows results of NSPSO with various Pn1 values. The Pn1 value at 60% half of the RMSE values achieved best results. Fig.8 presented comparison results of six algorithms.

Table 2. Root mean square error valuesareobtained by DPSO with various P_{r1} values. The best values among the four groups of parameters are shown in bold.

SL No.	Dataset Name	P_{r1}=0% RMSE	P_{r1}=30% RMSE	P_{r1}=60% RMSE	P_{r1}=90% RMSE
1	ALBRECHT	0.104665	0.133014	**0.127667**	0.153807
2	CHINA	0.003790	**0.003704**	0.003790	0.003797

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

3	COCOMO 81	0.141468	0.152859	**0.136135**	0.148602
4	DESHARNAIS	0.068093	0.055096	**0.055497**	0.077512
5	ISBSG	0.334304	0.346257	**0.330073**	0.326615
6	MAXWELL	**0.136350**	0.156296	0.156372	0.145702
7	MIYAZAKI94	**0.142869**	0.145030	0.143488	0.149310

Table 3.Root mean square error values are obtained by NSPSO with various P_{n1}values. The best values among the four groups of parameters are shown in bold.

SL No.	Dataset Name	P_{n1}=20% RMSE	P_{n1}=40% RMSE	P_{n1}=60% RMSE	P_{n1}=80% RMSE
1	ALBRECHT	0.114865	0.125895	**0.112488**	0.119726
2	CHINA	0.003759	0.003772	**0.003727**	0.003753
3	COCOMO 81	0.133476	0.144940	0.140047	**0.123139**
4	DESHARNAIS	**0.056060**	0.058370	0.055930	0.062567
5	ISBSG	**0.310670**	0.331454	0.320264	0.318074
6	MAXWELL	0.135975	0.132487	**0.118838**	0.163082
7	MIYAZAKI94	0.131254	**0.118203**	0.119595	0.118495

Table 4.Root mean square error values are obtained by DPSONS-Kapproachwith various $P_{r1,}$ P_{n1} andK. The best valued among the nine group of parameters are shown in bold.

DATA SET	P_{r1}=0% P_{n1}=20% RSME	P_{r1}=0% P_{n1}=40% RSME	P_{r1}=0% P_{n1}=60% RSME	P_{r1}=30% P_{n1}=80% RSME	P_{r1}=60% P_{n1}=80% RSME	P_{r1}=90% P_{n1}=20% RSME	P_{r1}=90% P_{n1}=40% RSME	P_{r1}=90% P_{n1}=60% RSME	P_{r1}=90% P_{n1}=80% RSME
K= 1									
ALBRECHT	0.083050	0.160294	0.116085	0.189092	0.098134	0.084016	0.112136	0.174906	**0.110833**
CHINA	0.003796	0.003803	**0.003778**	0.003793	0.003796	0.003552	0.003797	0.003803	0.003792
COCOMO 81	0.151067	0.130205	0.149231	0.129420	0.153321	0.114687	0.114618	0.120237	**0.109153**
DESHARNAIS	**0.044139**	0.069077	0.066555	0.042802	0.060163	0.051814	0.065969	0.073202	0.060727
ISBSG	0.348091	0.350285	0.341186	**0.264409**	0.336133	0.348772	0.340175	0.343868	0.333045
MAXWELL	0.109933	**0.097561**	0.132162	0.182902	0.166586	0.111106	0.085637	0.155006	**0.104269**
MIYAZAKI94	0.148697	**0.082478**	0.142388	0.080241	0.096206	0.148580	0.148456	0.148890	**0.113759**
K=2									
ALBRECHT	0.131447	**0.083900**	0.163954	0.124192	0.171351	0.124320	0.171097	0.184911	**0.093153**
CHINA	0.003789	0.003796	**0.003785**	0.003798	0.003796	0.003711	0.003792	0.003798	0.003794
COCOMO 81	0.116351	**0.108008**	0.144307	0.151096	0.148693	0.123504	0.151708	0.151440	0.151601
DESHARNAIS	0.080328	0.066035	0.062023	0.051339	0.044183	0.052041	0.050423	**0.042828**	**0.055118**
ISBSG	0.343629	**0.321293**	0.346656	0.330817	0.300964	0.343670	0.345123	**0.325029**	0.337873
MAXWELL	0.142697	0.159741	0.183441	0.081845	0.191748	0.127601	0.113242	0.125142	**0.110541**
MIYAZAKI94	0.148482	0.151351	0.151975	0.148483	**0.125357**	**0.071966**	0.146313	0.118596	**0.142550**
K=3									
ALBRECHT	**0.107737**	0.131598	0.135853	0.088342	0.175697	0.133831	0.087762	0.150928	0.136822
CHINA	**0.003687**	0.003791	0.003787	0.003794	0.003798	0.003783	0.003755	0.003795	0.003729

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

COCOMO 81	0.151753	**0.128373**	0.151471	0.124672	0.135318	0.152505	0.151377	0.150208	**0.151371**
DESHARNAIS	**0.047585**	0.077990	0.054392	0.058970	0.055738	0.062516	0.077912	0.071824	**0.047766**
ISBSG	0.341796	0.344083	0.329742	0.336668	0.343592	0.343548	0.344492	0.338420	0.266379
MAXWELL	0.156151	0.187720	**0.085099**	0.191391	0.092011	0.155206	0.121908	0.127464	**0.102124**
MIYAZAKI94	**0.073987**	**0.094882**	**0.120480**	**0.106450**	**0.116233**	**0.126389**	**0.108865**	0.148430	0.142408
K=4									
ALBRECHT	0.132572	0.132572	0.164546	0.090789	0.153885	0.107417	0.139121	0.142167	**0.087629**
CHINA	0.003800	0.003800	0.003695	0.003785	0.003787	0.003772	0.003780	0.003789	**0.003797**
COCOMO 81	0.151460	0.151460	**0.115271**	0.116077	0.110858	0.151596	0.151419	0.152488	0.118285
DESHARNAIS	0.065028	0.065028	0.057654	**0.045760**	0.045364	0.073924	0.044725	0.054739	**0.045810**
ISBSG	**0.282960**	**0.282960**	0.339313	0.306452	0.343748	0.306334	0.356162	0.343496	0.353660
MAXWELL	0.157085	0.157085	0.166625	**0.125575**	0.175281	**0.134191**	0.142106	0.181737	0.169146
MIYAZAKI94	**0.064101**	**0.064101**	0.139974	0.149302	**0.062350**	**0.085190**	0.148469	0.134413	**0.099098**
K=5									
ALBRECHT	0.154853	0.155742	0.133840	0.117531	0.141201	**0.084796**	0.124245	**0.105809**	0.153464
CHINA	0.003792	0.003795	0.003800	0.003798	0.003762	0.003800	0.003798	0.003801	**0.003710**
COCOMO 81	0.154680	**0.105108**	0.173087	0.151425	0.164535	0.211613	0.151540	0.162763	0.130997
DESHARNAIS	0.075302	0.062670	0.066499	0.051642	**0.047313**	0.050509	0.056881	**0.049709**	0.076156
ISBSG	0.345309	0.343910	0.348858	0.343667	0.350756	**0.343321**	0.350029	0.344554	0.352117
MAXWELL	0.198682	0.115223	0.168179	**0.100328**	0.164122	0.136215	0.181281	0.189682	0.186702
MIYAZAKI94	**0.065987**	0.142033	0.150670	**0.130652**	0.192302	0.158946	0.147575	0.148831	0.161072

Table 5. The ranking obtained by Friedman test for DPSONS-K with various parameter settings.

K= 1

DATA SET	P_{r1}=0% P_{n1}=20%	P_{r1}=0% P_{n1}=40%	P_{r1}=0% P_{n1}=60%	P_{r1}=30% P_{n1}=80%	P_{r1}=60% P_{n1}=80%	P_{r1}=90% P_{n1}=20%	P_{r1}=90% P_{n1}=40%	P_{r1}=90% P_{n1}=60%	P_{r1}=90% P_{n1}=80%
ALBRECHT	6	3	8	4	5	1	2	7	9
CHINA	7	5	4	1	3	9	8	2	6
COCOMO 81	2	5	3	7	6	9	1	8	4
DESHARNAIS	7	4	8	2	5	6	3	9	1
ISBSG	7	4	5	9	6	8	3	2	1
MAXWELL	5	2	8	6	4	3	9	7	1
MIYAZAKI94	5	9	2	8	4	6	7	3	1
K=2									
ALBRECHT	2	3	5	9	7	4	6	8	1
CHINA	8	3	5	1	7	2	4	9	6
COCOMO 81	4	9	2	5	3	1	8	6	7
DESHARNAIS	8	3	9	1	4	6	5	2	7
ISBSG	5	9	2	7	1	6	8	3	4
MAXWELL	8	7	3	4	9	6	2	5	1
MIYAZAKI94	2	9	7	8	4	1	6	3	5
K=3									
ALBRECHT	3	4	6	2	9	5	1	8	7
CHINA	1	6	5	7	9	4	3	8	2
COCOMO 81	8	2	7	1	3	9	6	4	5
DESHARNAIS	1	9	3	5	4	6	8	7	2
ISBSG	5	8	2	3	7	6	9	4	1
MAXWELL	7	8	1	9	2	6	4	5	3
MIYAZAKI94	1	2	6	3	5	7	4	9	8

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

K=4									
ALBRECHT	5	1	9	3	8	4	6	7	2
CHINA	9	6	1	4	5	2	3	7	8
COCOMO 81	7	1	3	4	2	8	6	9	5
DESHARNAIS	7	9	6	3	2	8	1	5	4
ISBSG	1	2	5	4	7	3	9	6	8
MAXWELL	5	3	6	1	8	2	4	9	7
MIYAZAKI94	2	8	6	9	1	3	7	5	4
K=5									
ALBRECHT	6	3	8	4	5	1	2	7	9
CHINA	7	5	4	1	3	9	8	2	6
COCOMO 81	2	5	3	7	6	9	1	8	4
DESHARNAIS	7	4	8	2	5	6	3	9	1
ISBSG	7	4	5	9	6	8	3	2	1
MAXWELL	5	2	8	6	4	3	9	7	1
MIYAZAKI94	5	9	2	8	4	6	7	3	1

Table 6. Rankingobtained by Friedman test for DPSONS-Kapproachwith various parameter settings. The lowest value is shown in bold.

DPSONS-K	Rankings of K				
	K=1	**K=2**	**K=3**	**K=4**	**K=5**
P_{r1}=0%, P_{n1} =20%	5.57	5.29	3.71	5.14	5.29
P_{r1}=0%, P_{n1}=40%	4.57	6.14	5.57	4.29	6.86
P_{r1}=0%, P_{n1} =60%	5.43	4.71	4.29	5.14	3.43
P_{r1}=30%, P_{n1}=80%	5.29	5.00	4.29	4.00	5.71
P_{r1}=60%, P_{n1} =80%	4.71	5.00	5.57	4.71	5.00
P_{r1}=90%, P_{n1} =20%	6.00	3.71	6.14	4.29	4.00
P_{r1}=90%, P_{n1} =40%	4.71	5.57	5.00	5.14	5.29
P_{r1}=90%, P_{n1} =60%	5.43	5.14	6.43	6.86	3.57
P_{r1}=90%, P_{n1} =80%	**3.29**	4.43	4.00	5.43	5.86

Table 7. Root square mean error valuesareobtained by variants of PSO, and DPSONS-Kapproachwhere k=1

DATASET	PSO	PSOW	PSOCL	PSOWFIP	PSOU	DPSO	NSPSO	DPSONS-K
ALBRECHT	0.126714	0.120294	0.157886	0.148707	0.120381	0.153807	0.119726	**0.110833**
CHINA	0.003758	0.003757	0.003745	0.003721	0.003737	0.003797	0.003753	**0.003792**
COCOMO 81	0.139883	0.133175	0.148104	0.135230	0.120046	0.148602	0.123139	**0.109153**
DESHARNAIS	0.063407	0.059000	0.057112	0.068283	0.047033	0.077512	0.062567	**0.060727**
ISBSG	0.342864	0.325460	0.356404	0.328101	0.305599	0.326615	0.318074	**0.333045**
MAXWELL	0.122367	0.139063	0.175546	0.155739	0.125796	0.145702	0.163082	**0.104269**

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

MIYAZAKI94	0.111315	0.126626	0.115592	0.123146	0.104569	0.149310	0.118495	**0.113759**

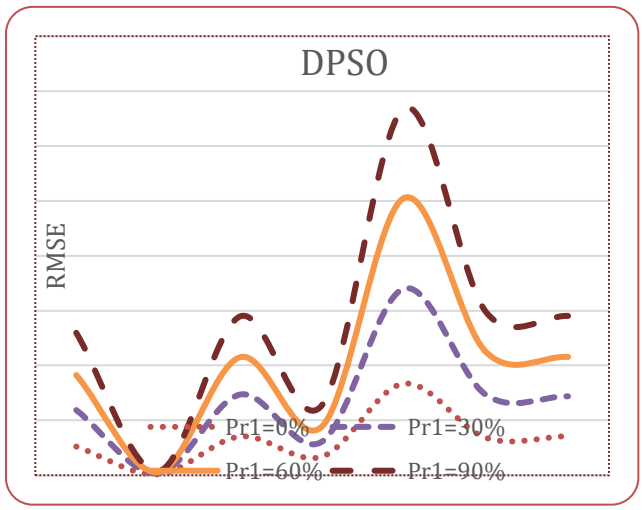

Figure.6 Comparison of RMSE values of DPSO with P_{r1} ofNSPSOwith Pn1

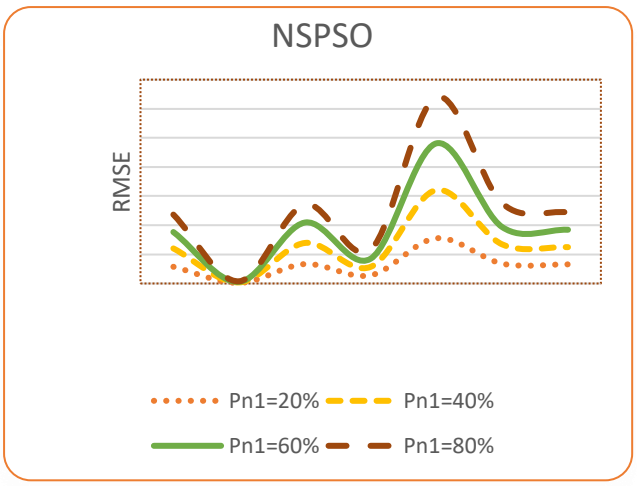

Figure.7 Comparison of RMSE values

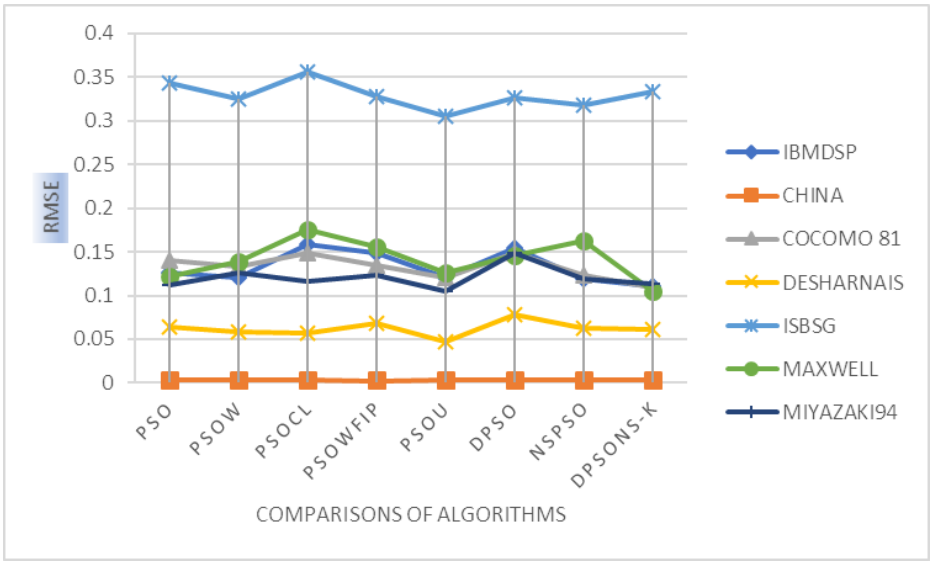

Figure 8 Comparison of RMSE values of DPSONS-K with other Algorithms

5. Conclusion

This paper presentinga novel hybrid algorithm is called DPSONS-Kmechanismto predict accurate software development cost. The DPSONS-K technique employs two methods are a diversity enhanced method and local and global neighborhood search with k tuning parameter. The earlier method is useful to enhance diversity between particles by creating a trail particle and reshuffle the particles. The later method is encouraging to accelerate convergence rate by tuning k in localand global neighbourhood search. By integrating these two methods, a hybrid mechanism is achieved balance between search and exploitation abilities. To validate performance of proposed method, make use of seven benchmark data sets. The parameters P_{r1}andP_{n1}may influence the performance of

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

proposed method. Experimental result shows that independent of either P_{r1}or $P_{n1}$are not suitable for accurate software development cost, but combination of these two parameters achieved best results.Either DPSO or NPSO cannot be achievingimpressive results. By combing them, the hybrid mechanism obtains better convergence rate.

References:

[1] S. Hastie and S. Wojewoda, "InfoQueue," Infoq, 4 October 2015. [Online]. Available: https://www.infoq.com/articles/standish-chaos-2015. [Accessed 4 June 2017].

[2] Qasim, Iqra, Hanny Tufail, Alia Fatima, Tayyaba Rasool, and Farooque Azam. "Cost estimation techniques for software development: A systematic literature review." In Proc. Int. Conf. Eng., Comput. Inf. Technol. (ICECIT), pp. 38-42. 2017.

[3] Muhammad Asif Saleem, Rehan Ahmad, Tahir Alyas, Muhammad Idrees, Asfandayar, Asif Farooq, Adnan Shahid Khan and Kahawaja Ali, "Systematic Literature Review of Identifying Issues in Software Cost Estimation Techniques" International Journal of Advanced Computer Science and Applications (IJACSA), 10(8), 2019.

[4] Sharma, T. N. "Analysis of software cost estimation using COCOMO II." International Journal of Scientific & Engineering Research 2, no. 6 (2011): 1-5.

[5] F. Marzoughi, M. M. Farhangian, A. Marzoughi and A. T. H. Sim, "A decision model for estimating the effort of software projectsusing Bayesian theory," 2010 2nd International Conference on Software Technology and Engineering, 2010, pp. V2-54- V2-59

[6] Chirra, Sai Mohan Reddy, and Hassan Reza. "A survey on software cost estimation techniques." Journal of Software Engineering and Applications 12, no. 06 (2019): 226.

[7] Kumar, Gaurav, and Pradeep Kumar Bhatia. "Automation of software cost estimation using neural network technique." International Journal of Computer Applications 98, no. 20 (2014).

[8] Gharehchopogh, FarhadSoleimanian, Isa Maleki, and Sahar Sadouni. "Artificial neural networks-based analysis of software cost estimation models." algorithms 20 (2014): 15.

[9] Mittal, Anish, Kamal Parkash, and Harish Mittal. "Software cost estimation using fuzzy logic." ACM SIGSOFT Software Engineering Notes 35, no. 1 (2010): 1-7.

[10] Kamal, Shahid, and Jamal Abdul Nasir. "A fuzzy logic-based software cost estimation model." International Journal of Software Engineering and Its Applications 7, no. 2 (2013): 7-18.

[11] Maleki, Isa, Laya Ebrahimi, SamanJodati, and Iraj Ramesh. "Analysis of software cost estimation using fuzzy logic." International Journal in Foundations of Computer Science & Technology (IJFCST) 4, no. 3 (2014): 27-41.

[12] Alaa F. Sheta, (2006), Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA Software Projects, Journal of Computer Science 2, (2) pp:118-123, 2006.

[13] Singh, Tribhuvan, Ranvijay Singh, and Krishn Kumar Mishra. "Software cost estimation using environmental adaptation method." Procedia computer science 143 (2018): 325-332.

[14] Oliveira, Adriano LI. "Estimation of software project effort with support vector regression." Neurocomputing 69, no. 13-15 (2006): 1749-1753.

[15] Raoofpanah, Hossein, and KhadijeHassanlou. "A probabilistic approach for project cost estimation using Bayesiannetworks." Life Science Journal 10, no. 6s (2013).

[16] Pahariya, J. S., Vadlamani Ravi, and MahilCarr. "Software cost estimation using computational intelligence techniques." In 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), pp. 849-854. IEEE, 2009.

[17] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.

[18] M. Dorigo, V. Maniezzo, A. Colorni, The ant system: optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics 26 (1996) 29–41.

[19] S.C. Chu, P.W. Tsai, Computational intelligence based on behaviors of cats, international journal of innovative computing, International Journal ofInnovative Computing, Information and Control 3 (1)(2007) 163–173.

[20] D. Karaboga, An Idea Based on Honey BEE Swarm for Numerical Optimization, Technical report TR06, Computer Engineering Department, Erciyes University, Turkey, 2005.

[21] Nayak, Sarat Chandra, and Bijan Bihari Misra. "Extreme learning with chemical reaction optimization for stock volatility prediction." Financial Innovation 6, no. 1 (2020): 1-23.

[22] Puspaningrum, Alifia, and RiyanartoSarno. "A hybrid cuckoo optimization and harmony search algorithm for software cost estimation." Procedia Computer Science 124 (2017): 461-469.

[23] Langsari, Kholed, and RiyanartoSarno. "Optimizing effort parameter of COCOMO II using

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

particle swarm optimization method." Telkomnika 16, no. 5 (2018): 2208-2216.
[24] Ahadi, Majid, and Ahmad Jafarian. "A new hybrid for software cost estimation using particle swarm optimization and differential evolution algorithms." Informatics Engineering, an International Journal (IEIJ) 4, no. 1 (2016).
[25] Padmaja, M., and D. Haritha. "Software effort estimation using grey relational analysis." MECS in International Journal of Information Technology and Computer Science 5 (2017): 52-60.
[26] Nassif, Ali Bou, Mohammad Azzeh, Ali Idri, and Alain Abran. "Software development effort estimation using regression fuzzy models." Computational intelligence and neuroscience 2019 (2019).
[27] Ullah, Aman, Bin Wang, Jinfan Sheng, Jun Long, Muhammad Asim, and Faiza Riaz. "A Novel Technique of Software Cost Estimation Using Flower Pollination Algorithm." In 2019 International Conference on Intelligent Computing, Automation and Systems (ICICAS), pp. 654-658. IEEE, 2019.
[28] Khazaiepoor, Mahdi, Amid KhatibiBardsiri, and FarshidKeynia. "A Hybrid Approach for Software Development Effort Estimation using Neural networks, Genetic Algorithm, Multiple Linear Regression and Imperialist Competitive Algorithm." International Journal of Nonlinear Analysis and Applications 11, no. 1 (2020): 207-224.
[29] Shahpar, Z., V. Khatibi, and A. KhatibiBardsiri. "Hybrid PSO-SA Approach for Feature Weighting in Analogy-Based Software Project Effort Estimation." Journal of AI and Data Mining 9, no. 3 (2021): 329-340.
[30] Venkataiah V., Ramakanta Mohanty, Nagaratna M.: Application of Practical Swarm Optimization to predict Software Cost Estimation. 6thIEEE International Conference on Communication Systems and Network Technologies, 05-07, March (2016).
[31] Eberhart, Russell, and James Kennedy. "Particle swarm optimization." In Proceedings of the IEEE international conference on neural networks, vol. 4, pp. 1942-1948. 1995.
[32] Garcia-Gonzalo, Esperanza, and Juan Luis Fernandez-Martinez. "A brief historical review of particle swarm optimization (PSO)." Journal of Bioinformatics and Intelligent Control 1, no. 1 (2012): 3-16.
[33] Shi, Yuhui. "Particle swarm optimization: developments, applications and resources." In Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546), vol. 1, pp. 81-86. IEEE, 2001.
[34] Wang, Hui, Hui Sun, Changhe Li, Shahryar Rahnamayan, and Jeng-shyang Pan. "Diversity enhancedparticle swarm optimization with neighborhood search." Information Sciences 223 (2013): 119-135.
[35] Shi, Yuhui, and Russell Eberhart. "A modified particle swarm optimizer." In 1998 IEEE internationalconference on evolutionary computation proceedings. IEEE world congress on computational intelligencepp. 69-73. IEEE, 1998.
[36] Clerc, Maurice, and James Kennedy. "The particle swarm-explosion, stability, and convergence in a multidimensional complex space." IEEE transactions on Evolutionary Computation 6, no. 1 (2002):58-73.
[37] Lim, Shi Yao, Mohammad Montakhab, and Hassan Nouri. "Economic dispatch of power system using particle swarm optimization with constriction factor." International Journal of Innovations in Energy Systems and Power 4, no. 2 (2009).
[38] Eberhart, Russ C., and Yuhui Shi. "Comparing inertia weights and constriction factors in particle swarmoptimization." In Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No 00TH8512), vol. 1, pp. 84-88. IEEE, 2000.
[39] Van den Bergh, Frans, and Andries Petrus Engelbrecht. "A study of particle swarm optimizationparticle trajectories." Information sciences 176, no. 8 (2006): 937-971.
[40] J. Kennedy, Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance, in: Proceedings of IEEE Congress on Evolutionary Computation, 1999, pp. 1391–1938.
[41] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, in: Proceedings of the Congress Evolutionary Computer, 1998, pp. 69–73.
[42] Suganthan, P.N. Particle swarm optimizer with Neighborhood operator. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; pp. 1958–1962.[43] X. Hu, R.C. Eberhart, Multiobjective optimization using dynamic neighborhood particleswarm optimization, in: Proceedings of the Congress Evolutionary Computer, 2002, pp. 1677–1681.
[44] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: simpler, maybe better,

IEEE Transactions on Evolutionary Computation 8 (3) (2004) 204–210.
[45] Mohais, Arvind S., Rui Mendes, Christopher Ward, and Christian Postoff. "Neighborhood re-structuring in particle swarm optimization." In Australasian Joint Conference on Artificial Intelligence, pp. 776-785. Springer, Berlin, Heidelberg, 2005.
[46] F. van den Bergh, A.P. Engelbrecht, A cooperative approach to particle swarm optimization, IEEE Transactions on Evolutionary Computation 8 (3) (2004) 225–239
[47] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Transactions on Evolutionary Computation 10 (2006) 281–295
[48] Y.P. Chen, W.C. Peng, M.C. Jian, Particle swarm optimization with recombination and dynamic linkage discovery, IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics 37 (6) (2007) 1460–1470.
[49] Hsieh, Sheng-Ta, Tsung-Ying Sun, Chan-Cheng Liu, and Shang-Jeng Tsai. "Efficient population utilization strategy for particle swarm optimizer." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39, no. 2 (2008): 444-456.
[50] A. Cervantes, I.M. Galván, P. Isasi, AMPSO: a new particle swarm method for nearest neighborhood classification, IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics 39 (5) (2009) 1082–1091.
[51] Z. Zhan, J. Zhang, Y. Li, H. Chung, Adaptive particle swarm optimization, IEEE Transactions on Systems, Man and Cybernetics– Part B: Cybernetics 39 (6) (2009) 1362–1381.
[52] Y. Wang, B. Li, T. Weise, J. Wang, B. Yuan, Q. Tian, Self-adaptive learning-based particle swarm optimization, Information Sciences 180 (20) (2011) 4515–4538
[53] W. Wang, H. Wang, S. Rahnamayan, improving comprehensive learning particle Swarm optimizer using generalized opposition-based learning, International Journal of Modelling, Identification and Control 14 (4) (2011) 310–316
[54] S. Das, A. Abraham, U. Chakraborty, A. Konar, Differential evolution using a neighborhood-based mutation operator, IEEE
Transactions on Evolutionary Computation 13 (3) (2009) 526–553.
[55] M. Pant, T. Radha, V.P. Singh. A simple iversity guided particle swarm optimization, in: Proceedings of IEEE Congress Evolutionary Computation, 2007, pp. 3294–3299
[56] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (1997) 341–359.
[57] Qingjian Ni, Jianming Deng, "Analysis of Population Diversity of Dynamic Probabilistic Particle Swarm Optimization Algorithms", Mathematical Problems in Engineering, vol. 2014, Article ID 762015, 9 pages, 2014
[58] Shi Cheng, Yuhui Shi, Quande Qin, "Dynamical exploitation space reduction in particle swarm optimization for solving large scale problems", Evolutionary Computation (CEC) 2012 IEEE Congress on, pp. 1-8, 2012
[59] Wang, Hui, Zhijian Wu, Shahryar Rahnamayan, Changhe Li, Sanyou Zeng, and Dazhi Jiang. "Particle swarm optimization with simple and efficient Neighborhood search strategies." International Journal of Innovative Computing and Applications 3, no. 2 (2011): 97-104.
[60] http://promise.site.uottawa.ca/SERepository
[61] https://zenodo.org/record/268446#.Ybwi6DNBy1s (china)
[62] http://promise.site.uottawa.ca/SERepository/datasets/cocomo81.arff
[63] http://promise.site.uottawa.ca/SERepository/datasets/desharnais.arff
[64] https://zenodo.org/record/268461#.YbwlbTNBy1s (Maxwell)
[65] https://zenodo.org/record/268465#.YbwlyDNBy1s (Miyazaki94)
[66] https://www.isbsg.org/tag/estimation