FOREX Publication
Open Access | Rapid and quality publishing

International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 2 | Pages 154-160 | e-ISSN: 2347-470X

Application of Chaotic Increasing Linear Inertia Weight and Diversity Improved Particle Swarm Optimization to Predict Accurate Software Cost Estimation

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

[1]Dept. of CSE , CMR College of Engineering & Technology, Hyderabad, India, venkat.vaadaala@gmail.com
[2]Dept. of CSE, JNTUH College of Engineering, Hyderabad, India, mratnajntu@gmail.com
[3]Dept. of CSE, Swami Vivekananda Institute of Technology, Hyderabad, India, ramakanta5a@gmail.com

*__Correspondence:__ V Venkataiah; Email: venkat.vaadaala@gmail.com

ABSTRACT- Nowadays usage of software products is increases exponential in different areas in society, accordingly, the development of software products as well increases by the software organizations, but they are unable to focus to predict effective techniques for planning resources, reliable design, and estimation of time, budget, and high quality at the preliminary phase of the development of the product lifecycle. Consequently, it delivered improper software products. Hence, a customer loses the money, time, and not belief on the company as well as effort of teamwork will be lost. We need an efficient and effective accurate effort estimation procedure. In the past, several authors have introduced different methods for effort of estimation of the software. Particle Swarm Optimization is a most popular optimization technique. Maintaining diversity in particle swarm optimization is the main challenging one and in this paper, we propose chaotic linear increasing inertia weight and diversity improved mechanism to enhance the diversity. Seven standard data sets were employed to analyze of performance of the proposed technique, and it is outperformed compared to the previous techniques.

Keywords: Software Cost Estimation (SCE), Particle Swarm Optimization (PSO), Software Effort Estimation (SEE), Root Mean Square Error (RMSE).

ARTICLE INFORMATION
Author(s): V Venkataiah, M Nagaratna and Ramakanta Mohanty
Special Issue Editor: Dr.S.Muthubalaji
Received: 09/04/2022; **Accepted**: 04/05/2022; **Published**: 30/05/2022;
e-ISSN: 2347-470X;
Paper Id: 0322SI-IJEER-2022-05
Citation: 10.37391/IJEER.100218
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-10/ijeer-100218.html

crossref member
CROSSREF.ORG
THE CITATION LINKING BACKBONE

This article belongs to the Special Issue on **Intervention of Electrical, Electronics & Communication Engineering in Sustainable Department**

Publisher's Note: FOREX Publication stays neutral with regard to Jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

SCE is a challenging task in the software product development community. It is process of predicting required resources to complete the process within a time limit and budget. Nowadays usage of software products is increases exponential in different areas in society, accordingly, the development of software products as well increases by the software organizations, but they are unable to focus to predict effective techniques for planning resources, reliable design, time, budget and required quality at the early stages of the software product development lifecycle. As a result, an inaccurate software product is delivered to the customers. Hence, the consumer loses money, time, and trust in a company. Various methods have been examined for SEE; including conventional methods may be the Cost Construct Model [4], and, supplementary machine learning techniques such as Neural Networks (NN) [1], bagging predictors [2], and Support Vector Regression (SVR) [3]. Machine learning approaches built a model for analyzing old projects, and to estimate new projects effort. For these techniques, huge amount of data samples are required to predict accurate results, but SCE estimation data sample size is very small. Hence, machine learning techniques give inaccurate results. Evolutionary techniques are Genetic Algorithms (GA), PSO, and more optimization techniques available in the literature. PSO has more popularity in the evolutionary community due to easy implementation, yet effectiveness, and its performance is greatly relevant to the diversity of the particles, especially when trying to avoid premature convergence and to get away from local optima. Hence, preserving a high diversity in PSO is a vital task.

In this paper, we propose combination of Chaotic Increasing Linear Inertia Weight and enhanced diversity technique of PSO to predict accurate Cost of the software at early stages of development. Remaining sections of the paper is structured in the following manner. Literature survey is presented in *Section 2*. In *Section 3* Employed Techniques. *Section 4* proposed mechanisms. Results and Discussions are presented in the *Section 5*. In *Section 6* paper conclusion.

2. LITERATURE SURVEY

The basic PSO was invented by Eberhart et al. in 1995 [5], it is a biological inspired evolutionary computation [6] and population-based stochastic search technique[7] start with an initial solutions of particles [8], required less memory, is computationally effective, and easier to implement[7]. For any search problem in Z-dimensional space, a particle can be represented by its position and velocity. Every particle is moved to its particle personal best (*pb*), and particle global best (gb) during an exploration process as followed [8].

$$u_{ij}(t+1) = iw.u_{ij}(t) + c_1.r_{1ij}.(pb_{ij}(t) - y_{ij}(t)) + c_2.r_{2ij}.(gb_{ij}(t) - y_{ij}(t)) \quad (1)$$

$$y_{ij}(t+1) = u_{ij}(t+1) + y_{ij}(t) \quad (2)$$

Where i= 1 to N is a population size and index of the particles, Yi = (y_{i1}, y_{i2}, ..., y_{iD}) is the ith particle position; Ui = (u_{i1}, u_i2, ..., u_{iD}) is the ith particle velocity; pb = (pb_{i1}, pb_{i2}, ..., pb_{iD}) is the ith particle of previous best position having better fitness values; and gb = (gb_{i1}, gb_{i2}, ..., gb_{iD}) is theparticle global best found from the overall particles until now. The parameter iw is inertia weight, which employed to balance between local and global particles search abilities [8].

PSO is a familiar optimizer and has been widely used in many practical applications since it was invented. Past thirty-year PSO has several variants. Starts with a summary of these different versions are can be presented as follows. Yuhui et al. [9] a special kind of parameter is called inertia weight iw embedded to the original PSO. Participation of iw is an important because of make a balance search and convergence activities, iw value decreasing linearly was a good choice for the search process. Constriction factors originally introduced by Clerc [10, 11, and 12] were used to solve different applications. Finally, all were decided was improves convergence velocity.

Bergh et al. [13, 15] highlighted in depth analysis of PSO parameters and provided mathematical evidence that each particle convenes to a check point. Kathrada [14] proposed flexible PSO (FlexiPSO) was including simple heuristics by which improves the flexibility of particle swarm. Lang et al. [16] introduced Comprehensive PSO (CPSO) which uses a novel learning mechanism is to gain knowledge from other particles in various dimensional to solve complex problems. Chen et al. [17] introduced another version of PSO was Dynamic Linkage Discovery to progress convergence rate of PSO, and is an effortless and successful linkage detection practice, and thereafter, a recombination operator that make use of dynamic linkage design to support the cooperation of PSO and Dynamic Linkage Discovery (DLD). Blackwell et al. [18] optimal values are obtained by dynamic function over the time period using a multi-swarm PSO. Li et al. [19] introduced adaptive learning PSO, which is a combination of its own personal best, nearest neighbor, and global best.

Khatibi et al. [20] proposed a hybrid mechanism which including PSO and analogy-based estimation to enhanced accuracy of SEE. Hassaniet al. [21] examined effects of the genetic algorithms, PSO algorithm and ant colony optimization, and presented comparative study of them. Shahpar et al. [22] proposed Polynomial Analogy Based Estimation (PABE) to improve the accurate prediction of the SCE. An analogy method is used to find out similarity property of the given project. Khatibi et al [23] proposed new service-oriented effort estimation is called global village service effort estimation which emphasizes an idea is think locally, operate globally. Khatibi et al. [24] propose a hybrid model which integrates Analogy Based Estimation (ABE) and Differential Evolution (DE) practice to predict software development services effort. Amid Khatibi Bardsiri [25] employed machine learning and statistical methods on different benchmark data sets to analyze data, build model and evaluate outcome. Venkataiah et al. [26] PSO to predict software cost estimation. K-means algorithms are used to frame the data of the cluster are given as input to PSO. The PSO is employed to determine the optimized result. But the major problem of PSO is suffering from the premature convergence means frequently struck at local minima.

In order to solve the diversity problem, number of methods was proposed. Standard PSO is used to test multi-model problems, which tend to suffer from the local minima and early convergence due to reduced diversity in a problem space. In order to overcome this problem of PSO, S. Das et al. [27] proposed the attractive and repulsive two stage mechanism in PSO is called (ARPSO). In stage one every particle is attracting each other because of information flow is good between particles, and hence the diversity decreases. In stage two individual particles are no longer attracted which means repelled by global best position and local best position. When the diversity drops under a lower bound dlower in stage one, ARPSO switches to the repulse stage, and diversity reached is high dhigher, then ARPSO switches back to attraction stage. The entire search process of ARPSO exchanges between attraction stage and repulsion stage- lower and higher diversities. However, the search behavior cannot be changed by ARPSO when diversity remains between dlower and dhigher. According to the ARPSO, Pant et al. [28] proposed a new stage is called the positive conflict stage as the middle stage between attraction stage and repulsion stage. In this stage, attraction, and repulsion are not complete present. Every particle is paying attention by its earlier best position and is repelled by the best global particle.

$$\text{Diversity} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\sum_{j=1}^{Z}\left(y_{ij}(t) - \acute{y}_{ıȷ}(t)\right)^2}, \quad (3)$$

$$\acute{y}_{ıȷ}(t) = \frac{\sum_{i=1}^{N} y_{ij}(t)}{N} \quad (4)$$

Where y'$_{ij}$ is the i^{th} position and j^{th} dimension of vector, N is entire particles that participated in the computation of diversity measurement and Z is the dimension vector. Qingjianet al. [29] proposed three measurements for improvement of diversity in PSO. Jonathan et al. [30] presented measures used to quantify diversity, and how they can be utilized to quantify the dispersion of both the candidate solutions and find out optima. Cheng et al. [31] Population diversity of PSO is vital when measuring and regulating algorithms ability of exploration or exploitation dynamically.

3. EMPLOYED TECHNIQUES

3.1 Chaotic Linear Increasing Inertia Weight

PSO performance basically rely on three components are called inertia weight, best personal and best global. Inertia weight is used to improve convergence rate of PSO. It has variant forms which were presented in [39]. In this paper, we propose chaotic linear increasing inertia weight. Chaotic inertia weight is derived from chaotic mapping. The Logistic mapping can be represented by $k = \mu \times k \times (1-k)$. When $3.57<\mu\leq 4$, is called chaotic phenomena. The result of chaotic phenomena is in closed interval of zero and one when $\mu=4$.

$$k = 4 * k * (1-k) \tag{5}$$

$$iw = w_{min} + \frac{(w_{max}+w_{min})}{maxiter} * iter * k \tag{6}$$

Where *iw* is inertia weight, w_{max} is maximum, and w_{min} is minimum inertia weights, maxiter is maximum iterations and iter is iteration.

3.2 Diversity Improved Method

To improve the diversity between particles, for swarm diversity maintained, we offer a new diversity improved method. For every particle $P1_i$ (t), a new particle $P1_i$ (t+1) is created by the PSO position updating equation. By using the following question creating a trail particle $P2_i$ (t+1) = (TY_i (t+1), TU_i (t+1))

$$TY_{ij}(t+1) = Y_{ij}(t+1) + (Y_{ij}(t+1) - Y_{ij}(t)).rand() \tag{7}$$

$$TU_{ij}(t+1) = U_{ij}(t+1) \tag{8}$$

Where *i*=1 to N, *j*=1 to D, and rand () is a random function which generate random value within range of (0, 1). Every element of the new position in a vector is X1i (t+1) obtained from the equation (5) and (6).

After, creating trail vector a greedy selection procedure is employed as follows:

$$P1_i(t+1)=\begin{cases} P2_i(t+1), & \text{if } f\left(P2_i(t+1)\right) \leq f\left(P1_i(t+1)\right), \\ P1_i(t+1), & otherwise \end{cases} \tag{9}$$

Where f (.) is evaluate fitness function, consider generality, this article only thinks about minimization problems. In addition to the new particle $P2_i$(t+1) is better than $P1_i$(t+1), then $P1_i$(t+1) replace with $P2_i$(t+1), otherwise, P1i(t+1) is remains same.

4. PROPOSED MECHANISM

In this paper, we proposed Chaotic Liner Increasing Inertia Weight and Diversity Improved Particle Swarm Optimization. Chaotic liner increasing inertia weight is used in a velocity update rule in each iteration generate to weight iw. DPSO which employs one strategy is particles are regularly moving same direction in a search process of PSO. It means that particles begin to be similar with the next iterations. When the trail P2i (t+1) is selected for next generation, the dissimilarities between P2i (t+1) and P1i (t+1) will find out, the dissimilarities among P1i (t+1), and remain particles of the swarm. High divergence in the swarm means high diversity and lower divergence in the swarm means lower diversity it leads to particles struck at local minima.

Proposed Algorithm

1. Normalize data;
2. Initialize every particle in the swarm randomly
3. Calculate effort according to COCOMO model refer to *Equation (10)*;
4. Determine the fitness value of Pi refer to *Equation (11);*
5. Initialize Pb and Gb
6. While Iteration <= MaxIterarions:
7. for *i* in range (ps):
8. Computation velocity of each particle $P1_i$ refer to *Equation (1)*;
9. Update position of each particle P1i refer to *Equation (2)*;
10. Set boundaries of position;
11. Determine the fitness value of P1i refer to *Equation (11);*
12. Create a new particle $X1_i$refer to *Equations (7) & (8)*;
13. Determine the fitness value of P2i refer to *Equation (11);*
14. Choose a check point between $P1_i$ and $P2_i$ as a new $P1_i$ refer *Equation (9)*
15. Update Pb_i and Gb_i;
16. End

All the necessary steps are given in *Proposed Algorithm*

5. RESULTS AND DISCUSSION

Seven datasets are used in experimentation to test the outcome of proposed method. Details about datasets were described in Table 1 with statistical information. These datasets are IBMDSP, CHINA, DESHARNAIS, MIYAZAKI, COCOMO81, ISBSG, and MAXWELL, and which are collected from the literature [31, 32, 33 34, 35, 36, and 37]. In all datasets, effort is considered as the targets attribute and remains of the attributes considered as input sources. Effort estimation can be calculated as follows.

$$\text{Effort (E)} = r*(\text{size})^{s} \tag{10}$$

Where effort can be measured in terms of PM (Person per month), r and s constants and size considered as Kilo Line of code. Min-Max normalization technique used to transfer larger data scale to small data scale because of to void the distortion in data. Normalized data is given as input to all proposed techniques. All the experimentation was conducted 20 times. *Table 2* describes root mean square error values achieved by the proposed technique on seven data sets. The Root Mean Square Error can be defined as

$$\text{RSME (E)} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(E_i - \bar{E}_i)^2} \tag{11}$$

Global optimums values were recorded. Where E_i old effort, $\bar{E}i$ new Effort, and n is a total number of data samples. For other parameters setting of PSO, population size N w depending on the dataset size, *iw* is chaotic increasing linear inertia weight, which is proposed in this paper, and c_1=1.48, c_2=1.50, total number of iterations is set 100.

Table 1. Describe Dataset details

SL No.	Dataset Name	Attributes	Observations	Minimum Effort	Maximum Effort	Average Effort	Standard Deviation	Unit
1	COCOMO 81	16	63	5.9	11400	683.32	1821.58	PM
2	IBMDSP	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

Table 2. RMSE values are achieved by proposed PSO. The best outcomes among four groups are shown in bold

Dataset	Inertia Weight iw=0.9	Proposed Inertia Weight iw	Proposed Diversity enhanced with 0.9	Proposed iw and Diversity enhanced
IBM DSP	0.105374	0.073157	0.113429	**0.073155**
CHINA	**0.003746**	**0.002947**	0.003792	**0.003611**
COCOMO 81	0.151454	0.112001	0.115622	**0.104874**
DESHARNAIS	0.043582	0.045542	0.045494	**0.040134**
ISBSG	0.343556	0.343543	0.343553	**0.343101**
MAXWELL	0.082313	0.081659	0.081646	**0.081638**
MIYAZAKI94	0.147936	0.062285	0.100285	**0.061731**

Table 3. Result Comparison of different inertia weight techniques

DATA SET	Linear Decreasing Inertia Weight	Chaotic Inertia Weight	Random Inertia Weight	Chaotic Random Inertia Weight	Proposed Inertia Weight
IBM DSP	0.081785	0.082301	0.076614	0.080086	**0.073155**
CHINA	0.003739	0.003793	0.003793	0.003794	**0.003611**
COCOMO 81	0.108440	0.107464	0.105096	0.104875	**0.104874**
DESHARNAIS	0.047484	0.047902	0.062093	0.061365	**0.040134**
ISBSG	0.343534	0.343545	0.343546	0.343610	**0.343101**
MAXWELL	0.081694	0.081673	0.088127	0.082504	**0.081638**
MIYAZAKI94	0.112073	0.111923	0.125640	0.125058	**0.061731**

FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 2 | Pages 154-160 | e-ISSN: 2347-470X

Table 4. Result Comparison of RMSE values by BASIC PSO, PSOW, PSOCF, UPSO, WFIPSO and Proposed. The best Results among six algorithms are presented in bold.

DATA SET	BASIC PSO	PSOW	PSOCF	UPSO	FIPSO	Proposed
IBM DSP	0.114065	0.104425	0.080037	0.094055	0.087083	**0.073155**
CHINA	0.003796	0.003795	0.003797	0.003795	0.003792	**0.003611**
COCOMO 81	0.152121	0.152305	0.127404	0.137457	0.127964	**0.104874**
DESHARNAIS	0.067600	0.063353	0.073989	0.054950	0.070214	**0.040134**
ISBSG	0.343468	0.345386	0.343665	0.280089	0.321057	**0.343101**
MAXWELL	0.105014	0.164904	0.134632	0.119220	0.176114	**0.081638**
MIYAZAKI94	0.149001	0.166394	0.141608	0.121141	0.156250	**0.061731**

According to the *Table 2* first column shows that performance PSO with inertia weight w=0.9 over seven data sets. Second column illustrate best results of PSO with proposed inertia weight when compared to first column. Third column presents results of proposed PSO diversity enhanced with inertia weight w=0.9 and last column shows that proposed mechanism which including proposed inertia weight and diversity enhanced, and the results of proposed mechanism is outperformed compared to remaining columns. Table 3describesanalysis of comparison results between various inertia weights are used in PSO on over seven datasets. Proposed inertia weight accomplished best results compared to the linear decreasing, chaotic, random, and chaotic random inertia weights. Table4 presents a comparative study on proposed method, and with other five variants of PSO on 7 datasets which are summarized in Table 1. Comparison of the proposed technique with PSO, PSOW, PSOCF, UPSO and FIPSO achieved better results than other techniques on all datasets except for PSOCF, and FIPSO on COCOMO 81. We used seven effort estimation benchmark datasets to analyze the performance of the proposed technique. Out of seven datasets, the CHINA dataset achieved the best global optimum value by the proposed technique which means that gap between actual and estimated effort should be minimized and followed by DESHARNAIS, IBM DSP, MIYAZAKI94, and MAXWELL. Similarly, FIPSO compared to PSO, PSOW, PSOCF, and UPSO achieved better results on only two datasets are IBM DSP and CHINA and did not give impressive results on the remaining of the datasets. Likewise, USPO achieved better results only on the ISBSG dataset. Equally, PSOCF achieved better results only on the COCOMO81 dataset. Performance of the PSO and PSOW is almost equally over seven datasets. Finally, *figure 1* Describes the root mean square error value over seven data sets by PSO, and proposed techniques. *Figure 2* shows the comparisons of PSO variants with root mean square values.

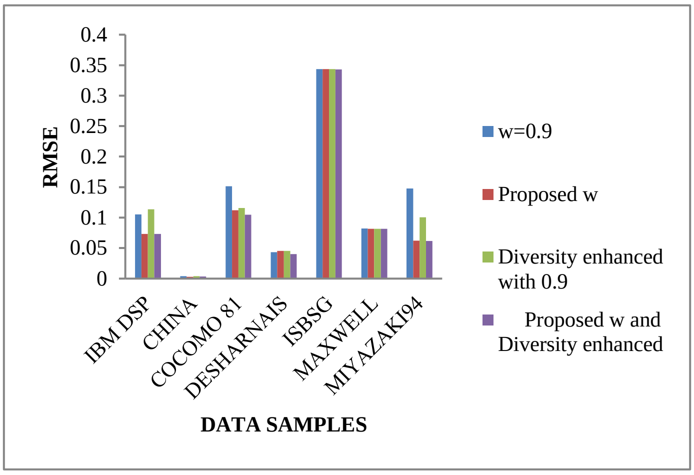

Figure 1: RMSE with various inertia weights

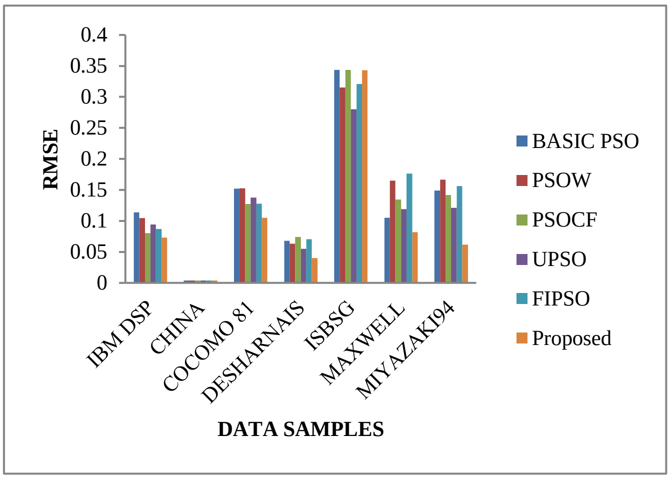

Figure 2: Root mean square error values vs six algorithms

6. CONCLUSON AND FUTURE WORK

SCE is a challenging activity in the product development Community. Estimating accurate product development costs at the early stages of the product process development is very difficult due to incomplete data. For accurate software cost estimation, this paper proposed chaotic linear increasing inertia weight and diversity improved PSO algorithm used to improve diversity between particles by search and convergence activists. To analyze the outcome of the proposed algorithm by using seven benchmark datasets. Result of proposed algorithm 3.12% better than standard PSO on IBM DSP dataset, 0.01% on CHINA, 4.74 % on COCOMO 81, 2.32% on DESHARNAIS, 0.22% on ISBSG, 8.32% on MAXWELL, and 10.46 % better than on MIYAZAKI94 dataset. Diversity improved PSO integrated with neighborhood search operators in future work.

REFERENCES

[1] Shin,M., Goel,A.L. (2000): Empirical data modeling in software engineering using radial basis functions. IEEE Transactions on Software Engineering, vol. 26, no. 6, pp.567-576.

[2] Oliveira, A. L. I. (2006): Estimation of software projects effort with support vector regression. Neuro computing vol.69, no. 13-15, pp. 1749-1753.

[3] Braga, P. L., Oliveira, A. L. I.; Ribeiro; G. H. T;and. MeiraS. R. L. (2007): Bagging predictors for estimation of software project effort. In: IEEE/INNS International Joint Conference on Neural Networks, IJCNN Orlando- Florida.

[4] Huang,X.; Ho, D., RenJ.; and CapretzL. F. (2007): Improving the COCOMO model using a neuro-fuzzy approach. Applied Soft Computing, vol. 7, pp. 29-40

[5] EberhartRussel, and James Kennedy (1995): Particle swarm optimization. In Proceedings of the IEEE international conference on neural networks, vol. 4, pp. 1942-1948.

[6] Garcia-Gonzalo, Esperanza, and Fernandez-MartinezJuan Luis ((2012). A brief historical review of particle swarm optimization (PSO). Journal of Bioinformatics and Intelligent Control 1, no. 1 pp. 3-16.

[7] Yuhui, Shi., (2001): Particle swarm optimization: developments, applications, and resources. In Proceedings of the 2001 congress on evolutionary computation (vol. 1, pp. 81-86.

[8] Wang, Hui., Hui Sun.;Change Li.; Shahryar Rahnamayan,; and Jeng-shyang Pan.(2013): Diversity enhanced particle swarm optimization with neighborhood search. Information Sciences 223 pp: 119-135.

[9] Shi, Yuhui, and Russell Eberhart.(1998): A modified particle swarm optimizer. In 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence pp. 69-73.

[10] Clerc; Maurice, and James Kennedy. (2002): The particle swarm-explosion, stability, and convergence in multidimensional complex space. IEEE transactions on Evolutionary Computation 6, no. 1 pp. 58-73.

[11] Lim; Shi Yao, Mohammad Montakhab; and Hassan Nouri (2009). Economic dispatch of power system using particle swarm optimization with constriction factor. International Journal of Innovations in Energy Systems and Power 4, no. 2

[12] Eberhart, Russ C, and Yuhui Shi. (2000): Comparing inertia weights and constriction factors in particle swarm optimization. In Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No 00TH8512), vol. 1, pp. 84-88.

[13] Van den Bergh, Frans, and Andries Petrus Engelbrecht (2010).: A convergence proof for the particle swarm optimizer. Fundamental Informatician 105, no. 4: pp. 341-374.

[14] Kathrada, M.: the Flexi-PSO (2009): Towards a more flexible particle swarm optimizer. SEARCH 46, 52–68.

[15] Van den Bergh,F., Engelbrecht A.P. (2004): A cooperative approach to particle swarm optimization, IEEE Transactions on Evolutionary Computation 8 (3) pp.225–239.

[16] Liang, J.J, QinA.K.;Suganthan,P.N, and Baskar,S. (2006): Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Transactions on Evolutionary Computation pp.281–295.

[17] Chen,Y.P. Peng, W.C.; Jian, M.C. (2007).:Particle swarm optimization with recombination and dynamic linkage discovery, IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics 37 (6) pp.1460–1470.

[18] Blackwell, T. Branke, J. (2004) Multi-Swarm Optimization in Dynamic Environments. In Workshops on Applications of Evolutionary Computation Springer: Berlin/Heidelberg, Germany, 2 pp. 489–500.

[19] Li,C.; Yang,S. (2009): An adaptive learning particle swarm optimizer for function optimization, in Proceedings of the Congress Evolutionary Computer, 2009, pp. 381–388.

[20] Khatibi Bardsiri, V., Jawawi, D.N.A., Hashim, S.Z.M. et al. A PSO-based model to increase the accuracy of software development effort estimation. Software Qual J 21, 501–526 (2013).

[21] Saadi, Maryam Hassani, Vahid KhatibiBardsiri, and Fahimeh Ziaaddini. "The application of meta-heuristic algorithms to improve the performance of software development effort estimation models." International Journal of Applied Evolutionary Computation (IJAEC) 6, no. 4 (2015): 39-68.

[22] Zahra Shahpar, Vahid Khatibi Bardsiri, Amid Khatibi Bardsiri: Polynomial analogy-based software development effort estimation using combined particle swarm optimization and simulated annealing. Concurr. Comput. Pract. Exp. 33(20) (2021)

[23] KhatibiBardsiri, FarshidKeynia. (2019) A Dataset-Independent Model for Estimating Software Development Effort Using Soft Computing Techniques. Applied Computer Systems 24:2, pages 82-93.

[24] KhatibiBardsiri, A., & Hashemi, S. M. (2016). A differential evolution-based model to estimate the software services development effort. Journal of Software: Evolution and Process, 28(1), 57-77.

[25] KhatibiBardsiri, Amid. "A new combinatorial framework for software services development effort estimation." International Journal of Computers and Applications 40, no. 1 (2018): 14-24.

[26] Venkataiah V., Ramakanta Mohanty, Nagaratna M (2019): Application of hybrid techniques to forecasting accurate software cost estimation. International Journal of Recent Technology and Engineering, ISSN: 2277-3878, Volume-7, Issue-6. Pp. 408-412

[27] Pant,M., Radha,T. andSingh.V.P. (2007): A simple diversity guided particle swarm optimization, in Proceedings of IEEE Congress Evolutionary Computation, pp. 3294–3299

[28] Storn,R, andPrice,K. (1997): Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 pp. 341–359.

[29] Qingjian Ni, and JianmingDeng. (2014).: Analysis of Population Diversity of Dynamic Probabilistic Particle Swarm Optimization Algorithms, Mathematical Problems in Engineering, vol. 2014, Article ID 762015, 9 pages.

[30] Mwaura,J, Engelbrecht,A.P.; and Nepomuceno, F.V.(2021)Diversity Measures for Niching Algorithms.

[31] Shi Cheng, Yuhui Shi, andQuande Qin. (2012).: Dynamical exploitation space reduction in particle swarm optimization for solving large scale problems, Evolutionary Computation (CEC) 2012 IEEE Congress on, pp. 1-8.

[32] Dr. J. S. Awati, Prof. S.S. Patil and Dr. M.S. Kumbhar (2021), Smart Heart Disease Detection using Particle Swarm Optimization and Support Vector Machine. IJEER 9(4), 120-124. DOI: 10.37391/IJEER.090405.

[33] http://promise.site.uottawa.ca/SERepository

[34] https://zenodo.org/record/268446#.Ybwi6DNBy1s (china)

[35] http://promise.site.uottawa.ca/SERepository/datasets/cocomo81.arff

[36] http://promise.site.uottawa.ca/SERepository/datasets/desharnais.arff

[37] https://zenodo.org/record/268461#.YbwlbTNBy1s (Maxwell)

[38] https://zenodo.org/record/268465#.YbwlyDNBy1s (Miyazaki94)

[39] https://www.isbsg.org/tag/estimation

[40] Bansal, Jagdish Chand, P. K. Singh, Mukesh Saraswat, Abhishek Verma, Shimpi Singh Jadon, and Ajith Abraham. "Inertia weight strategies in particle swarm optimization." In 2011 Third world congress on nature and biologically inspired computing, pp. 633-640. IEEE, 2011.