International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Optimizing Convergence and Diversity: A Synergistic Approach using Genetic Algorithm and Particle Swarm Optimization

Ashok Kumar
Associate Professor, Department of Computer Science and Engineering, GL Bajaj Group of Institutions, Mathura, (U.P)-India
Email: ashchhonkar@gmail.com

Anurag Kumar Singh
Assistant Professor, Department of Computer Science and Engineering GL Bajaj Group of Institutions, Mathura, (U.P)-India
Email: anuragsngh68@gmail.com

Sheo Kumar
Professor, Department of Computer Science and Engineering, CMR Engineering College, Hyderabad
Email: sheo2008@gmail.com

Abstract— PSO (Particle Swarm Optimization) is a widely used optimization technique. One notable variation of PSO is Auto Improved Particle Swarm Optimization (AIPSO). Due to its simplicity, PSO is widely used in many applications; nevertheless, although AIPSO converges quickly, it suffers from a noticeable stagnation problem. The Auto Improved Particle Swarm Optimization strategically upgrades itself by utilizing Genetic Algorithm (GA). This combination adds necessary variation, offsets the early convergence that is present in AIPSO, and successfully tackles the issue of stagnation. Combining AIPSO with Genetic Algorithm improves convergence and gets rid of stagnation problems. By taking use of the genetic algorithm's variety provision, the hybrid GA-AIPSO technique improves algorithm performance overall by preventing premature convergence to AIPSO-generated solutions. A detailed comparison with the most advanced algorithms, including JADE, GA-DE, and PGHA, confirms that the suggested hybrid GA-AIPSO is more successful. The algorithm's ability to break through stagnation and accelerate convergence rates in optimization problems is clearly demonstrated by the results.

Keywords—Particle Swarm Optimization; Genetic algorithm; Stagnation; Local Optima; Premature Convergence; Optimization.

I. INTRODUCTION

Genetic algorithms are randomized algorithms and widely used to solve complex and real life problems. It is developed by Holland and inspired by natural evolution [1, 2]. The GAs have been applied successfully in large number of fields such as fuzzy logic control, optimization design, expert systems, scheduling, neural networks, and many others[3-4].

An evolutionary computational model also contains Particle Swarm Optimization [PSO] which works on swarm intelligence. PSO is developed by Kennedy and Eberhart in 1995[5, 6]. Both GA and PSO are population based optimization Algorithms. This computational system is initialized with a population of random particles and the algorithm searches for optima by updating the velocities and positions of particles in upcoming generations.

Auto Improved Particle Swarm Optimization [AI-PSO], has been developed for improving the performance of PSO algorithm. However as AI-PSO is created by updating the formula used for PSO, it adhere the inherent limitations of PSO. Therefore AI-PSO may get stuck in local optimal solution due to the fast rate of information flow between particles. This leads particles with a loss in diversity that increases the possibility of being trapped in local optima. Another problem, which is associated with this algorithm, is the common problem of stochastic problem.

To remove these problems, we have integrated AIPSO with GA [1] algorithm. These algorithms are combined because GA provides necessary diversity to the solutions generated by AI-PSO and overcome the problem of stagnation. This novel algorithm GA-AIPSO has good convergence as it has ability to recover from stagnation problem by exploring all good regions in the search space. To create this hybrid version, GA is combined with AI-PSO so that the proposed algorithm can utilize the advantages of both the algorithms. GA-AIPSO performs well in both uni-modal and multimodal problems. It is able to maintain the diversity and does not converge in the local optima. It also improves the convergence rate of the algorithm.

To address the second issue which is related to parameter tuning, operators of GA and AI-PSO are tuned separately. As proper and fine tuning of the parameters may result in faster convergence of the algorithm, this tuning is very necessary. It may also remove the problem of stagnation which is vital for multimodal problems. At present, proper tuning of these parameters in GA and AI-PSO has been done automatically so that the algorithm does not face problem of stagnation and it should be able to capture multiple optimal solutions in multi model problems. The results of the proposed algorithm are compared with the latest version of existing algorithms such as particle swarm optimization (PSO bound),AIPSO,JADE etc on benchmark functions and the proposed algorithm prove its effectiveness and efficiency over the existing algorithms in most of the cases.

The rest of this paper is organized as follows. Section II explains about many hybrid PSO variants. The working of Genetic algorithm, basic PSO and its variant known AIPSO is explained in Section III. proposed novel GA-AIPSO hybrid variant is presented in section IV. Experimental setup and benchmark functions are covered in section V. Section VI includes comparison of proposed algorithm GA-AIPSO with existing hybrid PSO variants over benchmark functions mentioned in section V. Finally Section VII draws the conclusion.

II. Hybrid Variants of PSO

One of the hybrid algorithms that combine PSO [5, 6] and GA [1] is PSO GA-based hybrid algorithm (PGHA) [7]. In this algorithm, both PSO and GA are applied simultaneously to the population for a designated number of iterations; after this P individuals are selected from each of the two algorithms applied according to the probability. The individuals having larger fitness value will have more probability of getting selected. PGHA showed better results than both PSO and GA.
Another hybrid algorithm which was developed to solve a multi objective optimization problem is a hybrid DE (HDE) [8]. In addition to using DE [22] for explorative search, it adds a problem dependent local search to add exploitation factor to it. It updates the solution using Pareto dominance. One more algorithm that combines the features of Ant Colony Optimization (ACO) [23] and Genetic Algorithms (GA) is ACO-GA [9]. It improves the searching efficiency of the algorithms. GA applies its operators on complete population and ACO is more effective for local search. ACO-GA results in better performance. GA-DE combines the capabilities of GA and DE algorithms.

In GA-DE [10] proposed by Wen-Yi Lin, the crossover operation of real valued GA is modified according to the difference vector in Differential Evolution. The base vector is chosen as the base vector. In EU-GA-PSO [11], there is a combination of Euclidian distance based Genetic Algorithm and PSO. In this approach, the points at larger Euclidian distance are selected as mutation and crossover vectors to increase exploration.

One other hybrid algorithm is DE-PSO [12]. It combines the idea of taking difference vectors from DE with the changing velocity phenomenon of PSO. The performance of DE-PSO shows improvement in results when tested over BBOB functions.

III. GA and AIPSO Methods

I. A. *Genetic Algorithm*

To solve a problem, the Genetic Algorithm codes solution in the form of an individual chromosome. An initial population of individuals represents solutions of the problem. The solution space is represented by a distinct chromosome. At the time of search starts, few chromosomes are randomly chosen from the search space that forms the initial population. The computation of fitness of individuals is measured by an objective function. The operators such as selection, crossover and mutation are applied in sequence to obtain a new generation of chromosomes. This process is continuing until the termination criterion is met. Finally, obtain the best chromosome as a solution. The working of Genetic Algorithm is described as follow:

1) Random population of *n* chromosomes is generated.
2) The fitness *f(x)* of each chromosome is evaluated.
3) New population is created by repeating below steps until the new population is complete.
    a) Two parent chromosomes are selected from a population according to their fitness.
    b) Cross over the parents to form a new offspring applying crossover probability. If no crossover was performed, offspring is an exact copy of parents.
    c) Mutate new offspring at each locus with a mutation probability.
    d) New offspring is placed in a new population.
4) New generated population is used for running of algorithm.
5) If the number of populations or improvement of the best solution is satisfied, Stop, and return the best solution.
6) Go to step 2

B. *Auto Improved PSO (AI-PSO) Algorithm*

A number of agents or particles constitute a swarm. That moves around in the search space and finding for the best solution. Each particle is considered as a point in a D-dimensional space which adjusts itself according to its own flying experience as well as the flying experience of other particles. PSO maintains velocity vector $V_i = (v_i^1, v_i^2 \dots v_i^D)$ and position vector $X_i = (x_i^1, x_i^2 \dots x_i^D)$ for each ith particle of the population where D is dimension of search space.

The best solution (fitness) that has achieved so far by the particle by adjusting its coordinates in the solution space. This value of the best solution is called personal best, ***pbest***. Another best value that is obtained by the PSO is the best value obtained so far by any particle in the neighborhood of that particle. This value is called ***gbest***. Particles learn both from, their self best position attained previously (pbest) and best position attained by any particle in population so far (gbest). For the ith particle vector $pbest_i = (pbest_i^1, pbest_i^2 \dots pbest_i^D)$ represents best location obtained by it so far. The other vector $gbest = (gbest^1, gbest^2 \dots gbest^D)$ store best location gained by any particle among all particles. Velocity and position vectors of particles are updated in each

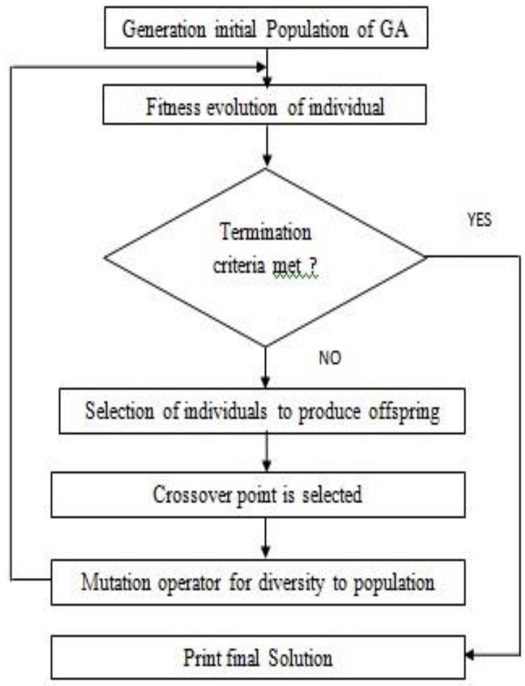

Generation initial Population of GA
Fitness evolution of individual
Termination criteria met ?
YES
NO
Selection of individuals to produce offspring
Crossover point is selected
Mutation operator for diversity to population
Print final Solution

Fig.1 Flow chart of GA

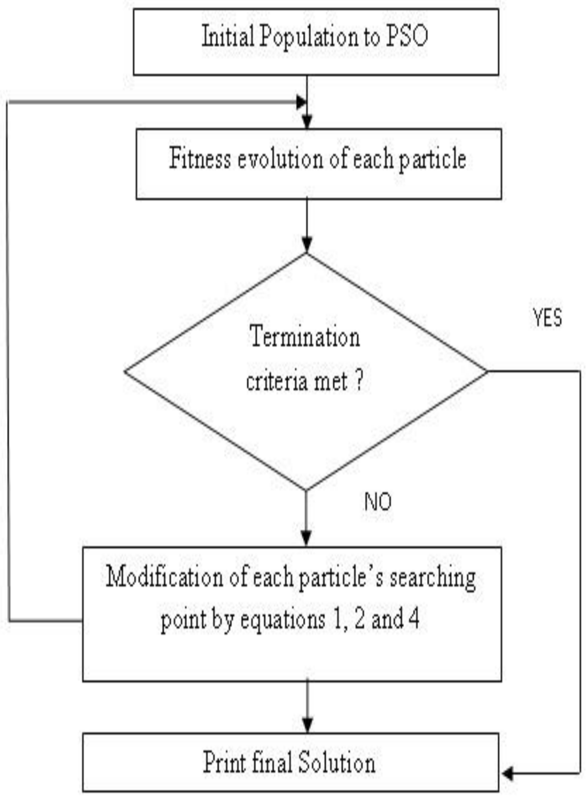

Initial Population to PSO
Fitness evolution of each particle
Termination criteria met ?
YES
NO
Modification of each particle's searching point by equations 1, 2 and 4
Print final Solution

Fig.2 Flow chart of AIPSO

iteration t (t = 1, 2, 3 …) using the equation (1) and (2) respectively:

$$V_i = \omega * V_i + c1 * r_1 * (pbest_i - X_i) + c2 * r_2 * (gbest - X_i) \quad (1)$$

$$X_i = X_i + V_i \quad (2)$$

Where, ω denotes inertia, c1 and c2 are the acceleration coefficients to control the relative contribution of $pbest_i$ and gbest respectively. The r_1 and r_2 are randomly generated numbers uniformly distributed in the range [0, 1].

The velocity equation is responsible to move all particles in direction of optimum solutions and it depends on three random parameters inertia, cognitive and social contribution. We have observed that the velocity operator of PSO need to be redefined because the operator which was used in PSO to update the velocity of particles is not effective for those particles whose value is equal to pbest and /or gbest . For all such particles whose values are equal to their pbest and /or gbest , the contribution of cognitive and/or social part for the solution will be zero. For example the particle who is at gbest will be motivated by the velocity Equation shown in equation (3).

$$V^k_{i\,(t+1)} = \omega * V^k_{i\,(t)} \quad (3)$$

Due to this velocity vector, there will be very little change in the position of this particle. Hence there will be a little change in the position of this particle. Similar situation will be faced by those particles in which at least one such component is zero. As these particles are already at the best position so exploitation will not work, they must be directed to search the entire search space and must be used for exploration.

This analysis reveals that searching ability of PSO algorithm can be further improved by redesigning the velocity operator of PSO algorithm in equation (1) . For such particles that are already at pbest and gbest we have updated the equation as follows.

$$V_i = \omega * V_i + c1 * r_1 * (pbest_i - X_i) + c2 * r_2 * (gbest - X_i) + r_3 * (gbest - pbest_i) \quad (4)$$

This additional factor will motivate these particles somewhere in the direction of gbest and pbest so that new good solutions can be identified.

Moreover to prevent each particle to converge on local optimum, following values of parameters ω , c1 and c2 are taken.

$$\omega = |G - Pavg| / |G| \quad (5)$$

$$c1 = |P_i - X_i| / |P_i| \quad (6)$$

$$c2 = |(G - X_i)| / G \quad (7)$$

Where, Pavg is the average of fitness of all pbest particles.|| is used to indicate the fitness of mentioned particles not the position particles. The steps involved in AI-PSO Algorithm are given as follows.

1) Position and velocity vector *X* and *V* are Initialized randomly for each particle of swarm

2) Initialized and generate values of random parameters using equation (5, 6 and 7).

3) *set itr* = 1

4) while function evaluations *(FEs) < Max FEs*

5) Evaluate the fitness of each particle
6) Update velocity vector *V* by using equation (1) and equation (4) as applicable
7) Update position vector *X* using equation (2)
8) *Increment itr = itr* + 1
9) End while

IV. IMPROVED HYBRID GA-AIPSO ALGORITH

The aim of this approach is to improve the AI-PSO algorithm by using the goodness of GA. Let we are performing crossover operator between two parents P1 and P2 having the fitness value f1 and f2 and each of length L (chromosome length). We assume that fitness of male is always greater than female and cut point is to be determined by Male. We will calculate the crossover point by calculating ((f1/(f1+f2))*L) and ((f2/(f1+f2))*L) (f1 is the fitness of male chromosome) and rounding off this to say variable C1 and C2. On the basis of this value C1, we will cut the parents at these sites and generate two children. The value C1 will always be less than L. Similarly we repeat the same process for C2 and generate two mutated children. We then perform binary tournament selection and select best mutant. Now we will show that application of this crossover and mutation will not affect randomness of genetic algorithm. As we are unaware about the nature of fitness value associated with each population, so fitness values will appear as random values associated with the parents and this operator will generate random values of children. So this operator will preserve the randomness of genetic algorithm.

Pesudocode for the process of this genetic algorithm is as follows:

STEP 1: Create an initial population N (P_0) and calculate the value of fitness for each chromosome.

STEP 2: Generate offspring population Q_t from P_t by using binary tournament selection, crossover and mutation operators.

STEP 3: Create offspring population Q_{t+1} from P_{t+1} by using the tournament selection, crossover and mutation operators.

STEP 3.1: Tournament Selection: For each population generated randomly, randomly pick two members from Q_t and a binary tournament is done between two based on fitness function f(x).Winners are further propagated for crossover.

STEP 3.2: Crossover: Two strings are crossed based on the crossover point. In proposed approach, crossover point= (f1(x)/f1(x) +f2(x))*Length (chromosome).

STEP 3.3: Mutation: point = (f2(x)/f1(x) +f2(x))*Length (chromosome). Perform binary tournament selection and pick best one.

STEP 4: This generational process is repeated until a termination condition has been reached.

AI-PSO algorithm may converge to a local optimum. We have utilized the operations of GA: selection, crossover and mutation to maintain diversity as well as for flying to new search area. This algorithm executes both the optimization technique in parallel and after combining ,sorting of are sorted on the basis of fitness function .If the individual from the AI-PSO has the fitness greater than the individual from the GA, then that particular individual from the GA is replaced by the individual from the AI-PSO. Thus the population formed will be continued for the next iteration. This gives a population in which the individuals are highly fit for the next generation. Finally if the termination criterion is met, the algorithm states the final solution.

The step-wise description of our implementation is described below:

- Initially, assign random population to AI-PSO.
- Apply GA to same population.
- Run the first iteration with GA as well as AI-PSO.
- Sort the outputs of both according to their fitness.
- Merge the outputs into a new array.
- Improve the first half of the chromosomes using GA.
- Improve the rest half using AI-PSO taking pbest and gbest of the first half best values.

Repeat steps 4-7 until stopping criteria (maximum number of iterations).

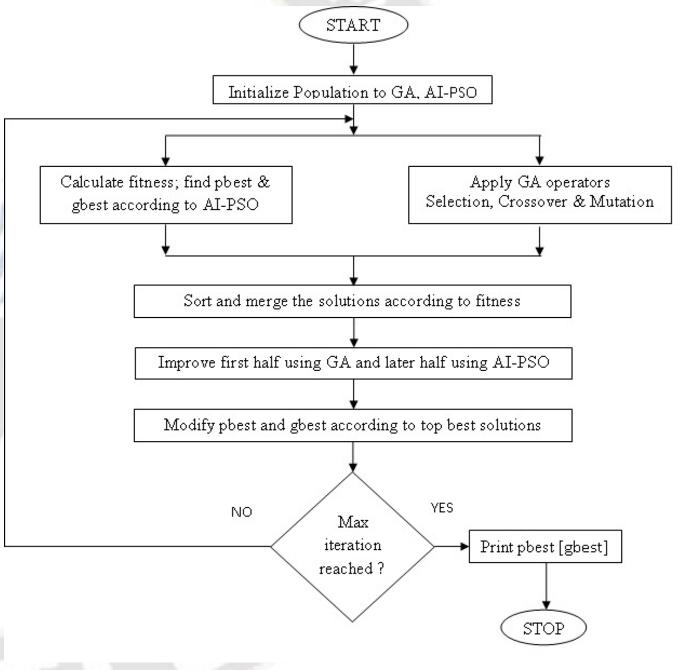

START
Initialize Population to GA, AI-PSO
Calculate fitness; find pbest & gbest according to AI-PSO
Apply GA operators Selection, Crossover & Mutation
Sort and merge the solutions according to fitness
Improve first half using GA and later half using AI-PSO
Modify pbest and gbest according to top best solutions
NO
Max iteration reached ?
YES
Print pbest [gbest]
STOP

Fig.3 Flow chart of GA-AIPSO

V. TEST FUNCTIONS AND EXPERIMENTAL SETUP

The performance of proposed GA-AIPSO algorithm is tested using Black-Box Optimization Benchmarking (BBOB) functions [13] in the experiments. using COmparing Continuous Optimizers (COCO) software [14, 15], GA-AIPSO is compared with AIPSO and its existing variants over BBOB benchmark functions.

A. Benchmark Functions

24 noise-free real-parameter single-objective benchmark functions are provided by The Black-Box Optimization Benchmarking (BBOB) [15, 16]. These functions are divided into five groups as below:

- Separable functions (f1 - f5)
- Functions with low or moderate conditioning (f6 - f9)
- Unimodal functions with high conditioning (f10 - f15)
- Multi-modal functions with adequate global structure (f16 - f19) and
- Multi-modal functions with weak global structure (f20 - f24).

All above benchmark functions are scalable with the dimension (D). Most functions have no specific value of their optimal solution i.e. they are randomly shifted in x-space. All functions have an artificially chosen optimal function value i.e. they are randomly shifted in f-space [31]. The search space for separable functions can be reduced to D -dimensional search procedures. The functions of first group (f1 - f5) are separable and rest benchmarks functions are non-separable.

B. Experimental Setup

The search domain for all functions is given as $[-5, 5]^D$. In the experiments all functions are tested for D = 2, 3, 5, 10 and 20 search space dimensionalities. If f_{opt} is optimal function value, defined for each benchmark function individually and Δf is required precision i.e. tolerable difference to optimal function value then $f_{target} = f_{opt} + \Delta f$ is target value of function to reach. COCO framework uses the value of Δf as 10^{-8} in computations. The number of trials run for each function per dimension is 15. Maximum allowed function evaluations (FEs) for each trial are calculated using the equation (8)

$$\text{Maximum\_FEs} = 1000 * D * \text{Population\_Size} \qquad (8)$$

where,

$$\text{Population\_Size} = \begin{cases} 20 * \text{Dimension} & \text{for } D = 2, 3 \\ 15 * \text{Dimension} & \text{for } D = 5 \\ 10 * \text{Dimension} & \text{for } D = 10, 20 \end{cases} \qquad (9)$$

VI. Experimental Results and Discussion

The performance of the proposed algorithm is evaluated on COCO framework [13, 14] on BBOB benchmark functions [15, 16]. The proposed algorithm is compared with 10 other state-of-art algorithms that are JADE [17], EU-GA-PSO [11], CMA-ES with increasing population size for the maximal number of 400(D+2) function evaluations (IPOP400D) [18], PGHA [7], HDE [8], ACO-GA [9], GA-DE [10], CMA-ES with Gaussian processes version 1 (GP1-CMAES) [19], DE-PSO [12], Genetic Algorithm with population size 100 (GA-100) [20]. Dataset of these algorithms are taken from [21].

Results from experiments are presented in figure number 4 to 6. The expected running time (ERT) is used in the figures. ERT depends on a given target function value, $f_t = f_{opt} + \Delta f$, and is computed over all relevant trials as the number of function evaluations executed during each trial while the best function value did not reach ft, summed over all trials and divided by the number of trials that actually reached it. Statistical significance is tested with the rank-sum test for a given target Δf_t using, for each trial, either the number of needed function evaluations to reach Δf_t (inverted and multiplied by -1), or, if the target was not reached, the best Δf-value achieved, measured only up to the smallest number of overall function evaluations for any unsuccessful trial under consideration if available.

A Empirical cumulative distribution functions (ECDF)

Empirical cumulative distribution functions (ECDF), plotting the fraction of trials with an outcome not larger than the respective value on the x-axis. Fig. 4 (i) and (iii) include ECDF of the number of function evaluations (FEvals) divided by search space dimension D, to fall below $f_{opt} + \Delta f$ with $\Delta f = 10^k$, where k is the first value in the legend. The thick red line represents the most difficult target value $f_{opt} + 10^{-8}$.

Legends indicate for each target the number of functions that were solved in at least one trial within the displayed budget. Fig. 4 (ii) and (iv) include ECDF of the best achieved Δf for running times of 0.5D, 1.2D, 3D, 10D, 100D, 1000D,.. … function evaluations (from right to left) and final Δf-value (red), where Δf and Df denote the difference to the optimal function value. Light brown lines in the background show ECDFs for the most difficult target of all algorithms benchmarked during BBOB-2009. The top row shows results for 5-D and the bottom row for 20-D.

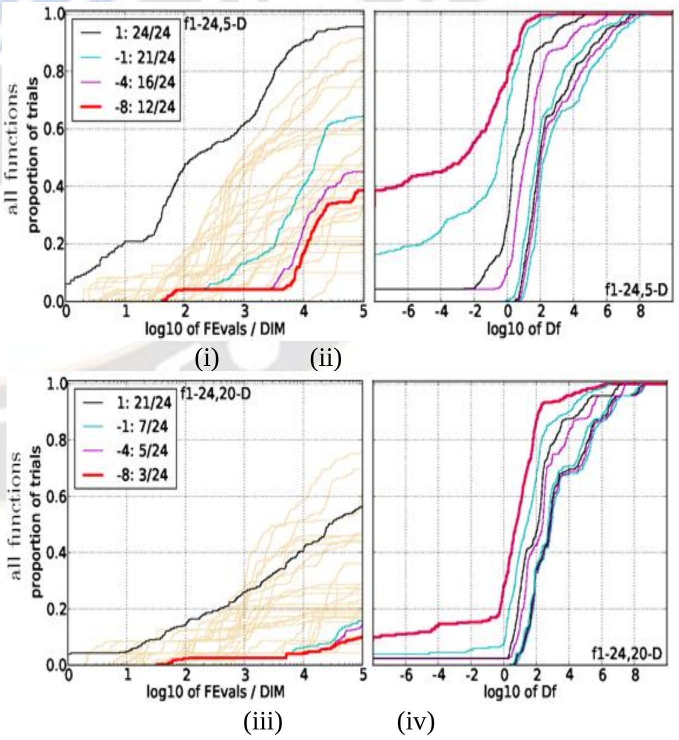
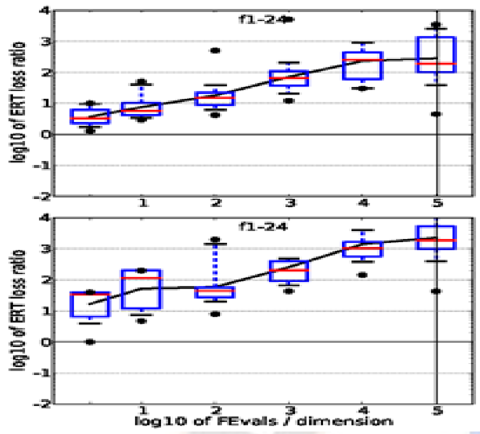


Fig.4. Plotting is fraction of trials versus function evaluations/dimesion or versus Df. (i) - (ii) for 5-D & (iii) - (iv) for 20-D.

B. ERT Loss Ratio

Fig. 5 display plotted versus given budget FEvals = #FEs in log-log. Box-Whisker plot shows 25-75%-ile (box) with median, 10- 90%-ile (caps), and minimum and maximum ERT loss ratio (points). The black line is the geometric mean. The vertical line gives the maximal number of function evaluations. And tabulated ERT loss ratios in 5-D and 20-D respectively. maxFE/D gives the maximum number of function evaluations divided by the dimension. RLUS/D gives the median number of function evaluations for unsuccessful trials.



$f1\text{–}f24$ **in 5-D**, maxFE/D=104902

#FEs/D	best	10%	25%	**med**	75%	90%
2	1.3	1.6	2.3	3.3	6.3	10
10	3.0	3.3	4.1	5.8	11	50
100	4.2	5.9	8.9	15	23	88
1e3	12	20	35	66	1.2e2	2.3e2
1e4	30	31	60	2.5e2	4.5e2	1.2e3
1e5	4.5	28	98	1.9e2	1.4e3	2.8e3
1e6	4.5	28	1.2e2	5.5e2	5.8e3	1.4e4
RL_{US}/D	1e5	1e5	1e5	1e5	1e5	1e5

$f1\text{–}f24$ **in 20-D**, maxFE/D=100537

#FEs/D	best	10%	25%	**med**	75%	90%
2	1.0	3.8	6.4	33	40	40
10	4.8	7.4	12	1.2e2	2.0e2	2.0e2
100	7.9	17	28	44	61	2.0e3
1e3	44	65	88	2.0e2	4.0e2	2.4e3
1e4	1.4e2	3.7e2	5.3e2	1.1e3	1.8e3	2.4e4
1e5	43	3.1e2	9.8e2	1.9e3	5.4e3	1.4e4
1e6	1.2e2	1.1e3	2.7e3	8.2e3	1.7e4	2.9e4
RL_{US}/D	1e5	1e5	1e5	1e5	1e5	1e5

Fig. 5 ERT loss ratio for 5-D and 20- D

*Expected nmber **of function** evaluations*

Expected number of function-evaluations (ERT, lines) to reach $f_{opt} + \Delta f$ is shown in fig. 6. Notation '+' is used to show Median number of f-evaluations to reach the most difficult target that was reached not always but at least once.

Notation 'x' is used to show Maximum number of function evaluations in any trial.

All values are divided by dimension and plotted as log10 values versus dimension. $\Delta f = 10^{\{1, 0, -1, -2, -3, -5, -8\}}$. Numbers above ERT-symbols (if appearing) indicate the number of trials reaching the respective target. The light thick line with diamonds indicates the respective best result BBOB 2009 for $\Delta f = 10^{-8}$.

Different symbols correspond to different algorithms given in the legend of f1 and f24. Light symbols give the maximum number of function evaluations from the longest trial divided by dimension. Horizontal lines give linear scaling, slanted dotted lines give quadratic scaling. Black stars indicate statistically better result compared to all other algorithms with p < 0:01 and Bonferroni correction number of dimensions (six)

: ◇ : JADE, ○ :GA-AIPSO, ✳ : EU-GA-PSO , ⬡ GA-DE, □ :PGHA, △: DE-PSO alba noiseless ◁ :HDE, ⬡: GP1-CMAES ◇ :GA-100, GA-DE, ▽ :AI-PSO, ⬠ :IPOP400D, Y :ACO-GA .

IV Comparison of Convergence Rate & Solution Accuracy

The performance of proposed algorithm is compared with peer algorithms for convergence rate and solution accuracy. Fig.7 - 8 shows the comparison of PSO variants for 5-D and 20-D.

Result on 5-D: the overall performance of GA-AIPSO is better than other peer algorithms on 5-Dimension as shown in Fig.7. For separable, Low or moderate conditioning and unimodal with high conditioning functions Algorithms JADE,PGHA and EU-GA-PSO converge faster than others. For multi-modal with adequate global structure and multi-modal with weak global structure GA-AIPSO perform best compare with other functions. In terms of solution accuracy, GA-AIPSO achieves optimal solution for all 24 benchmark functions.

Result on 20-D: GA-AIPSO performs significantly better than in multimodal problems (f15 - f24) as shown in fig. 8. As finding global optimal solution for multimodal problems is more difficult, GA-AIPSO performs comparatively superior than other algorithms.

VII. Conclusion-The experimental results demonstrate a significant improvement in AI-PSO's performance compared to both the traditional PSO and its many variations. In an effort to boost the AI-PSO algorithm's effectiveness even further, a new hybrid iteration called GA-AIPSO is presented. The goal of this hybrid model is to exploit the advantages of both Genetic Algorithm (GA) and AI-PSO in a synergistic manner. The subsequent analysis, carried out on a variety of benchmark functions including both unimodal and multimodal situations inside the COCO framework, carefully examines GA-AIPSO's performance. *The comparative analysis extends its purview to include the latest state-of-the-art algorithms, and the results unequivocally unveil a pronounced elevation in the performance metrics of the proposed hybrid algorithm, GA-AIPSO. This surge in performance is particularly evident across diverse dimensions, establishing GA-AIPSO as a robust contender in the optimization landscape. The findings thereby highlight not just an increase but a substantial and competitive enhancement in the performance of GA-AIPSO, positioning it favorably alongside leading algorithms in the field.*

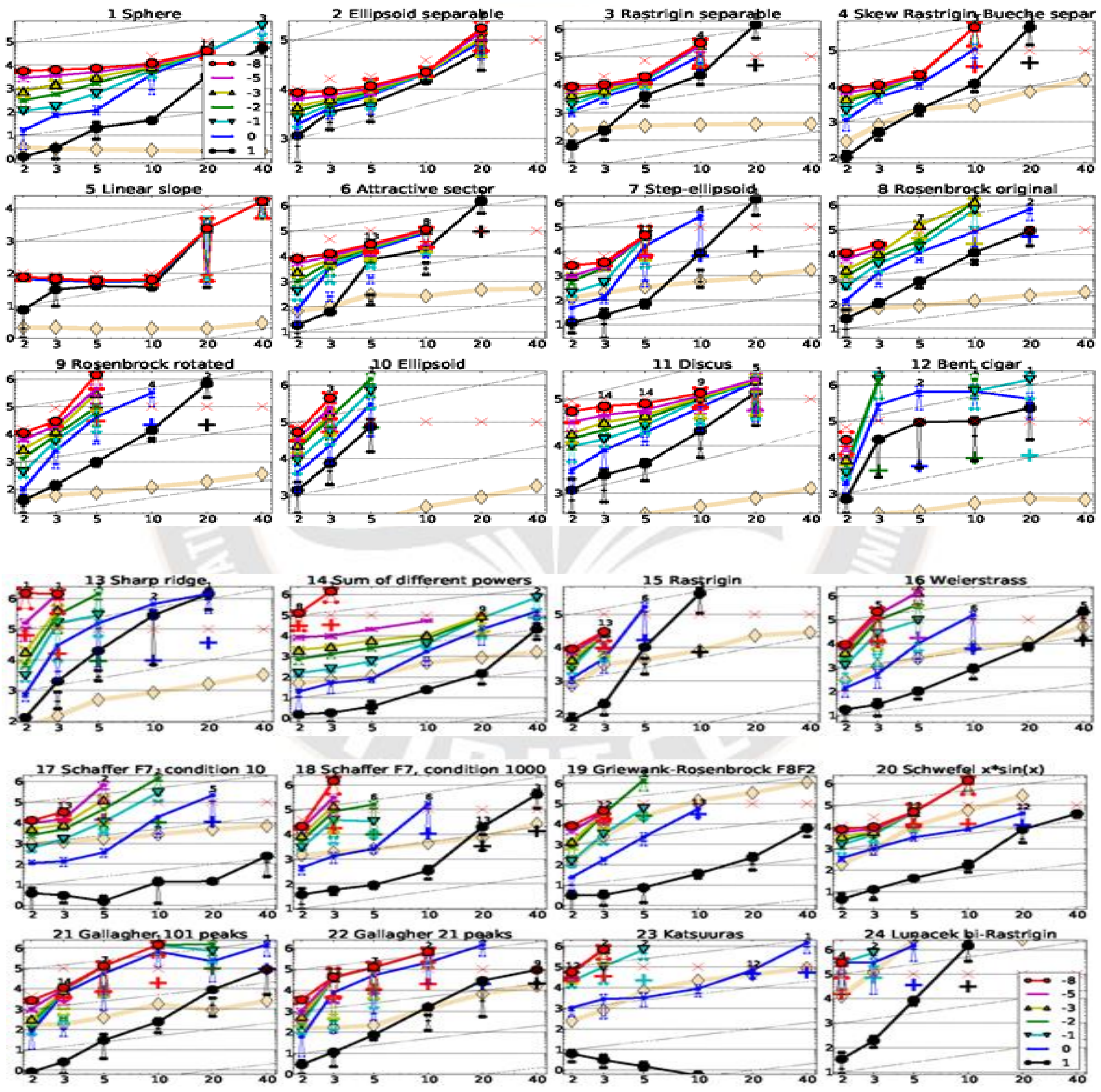


Fig. 6. Expected number of function evalution

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

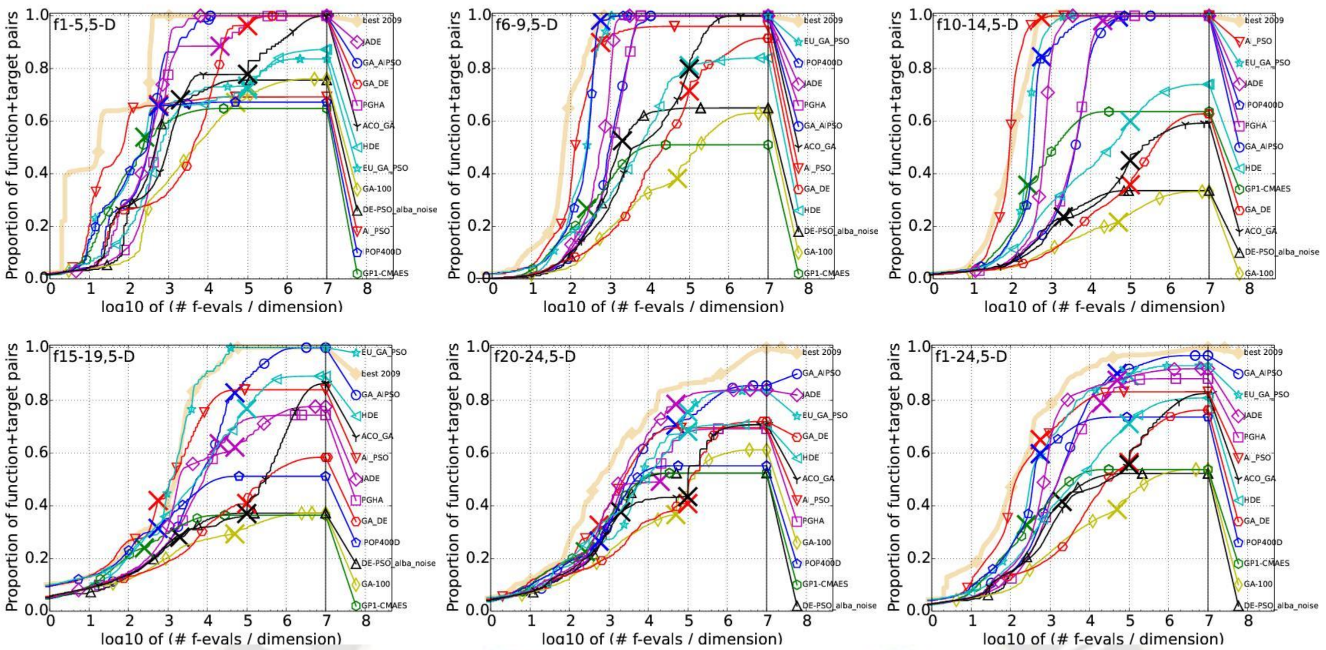


Fig.7 Convergence graph on 5-Dimension for BBOB benchmark functions (i) functions(f1-f5) (ii) functions(f6-f9) (iii) functions(f10-f15) (iv) functions(f16-f19) (v) functions(f20-f24) (vi) functions(f1-f24)

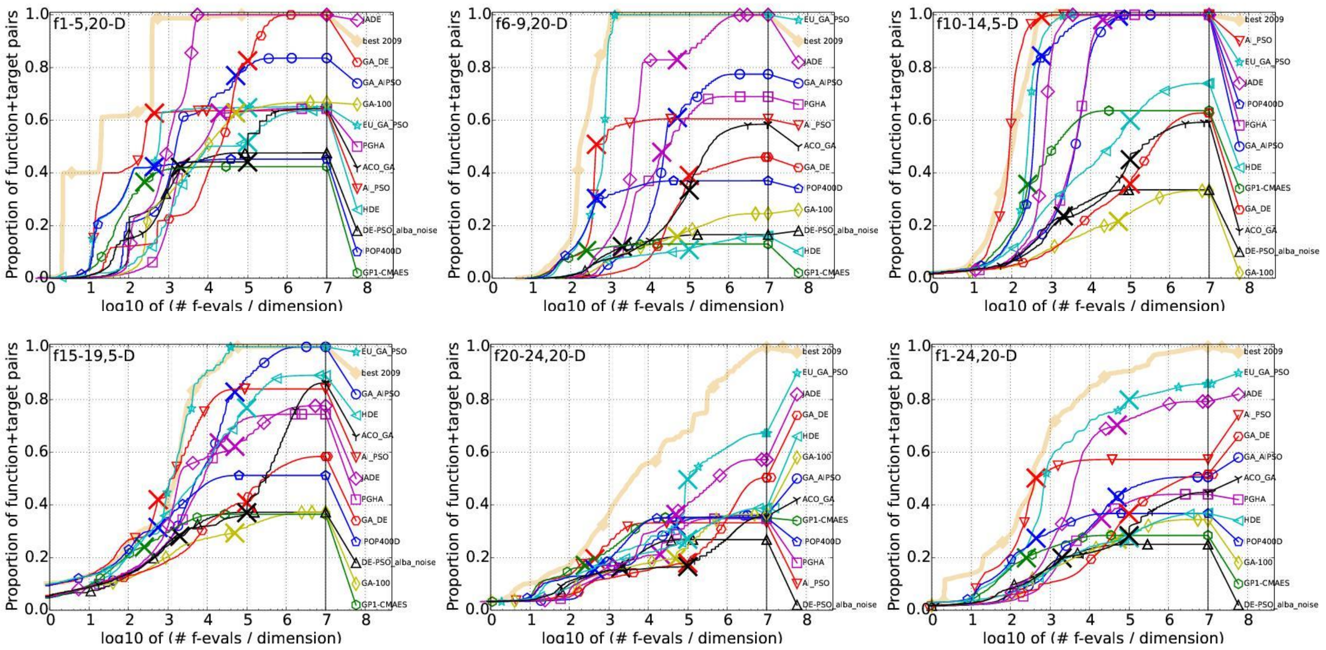


Fig.8 Convergence graph on 20-Dimension for BBOB benchmark functions (i) functions(f1-f5) (ii) functions(f6-f9) (iii) functions(f10-f15) (iv) functions(f16-f19) (v) functions(f20-f24) (vi) functions(f1-f24)

References

1. Holland, J. H. "Adaptation in Natural and Artificial Systems, the University of Michigan Press, Ann Arbor, MI. 1975." (1975).
2. Goldberg,D.E. "Genetic Algorithms inSsearch,Optimization & machine learning",Reading MA: Addison-Wesley,1989.
3. Stender,Joachim." Parallel Genetic Algorithms:Theory and Applications",[M] IOS Press,1993.
4. Man,K.F.,Tang,K.S.,Kwong,S.,"Genetic Algorithms: Concepts and Applications",IEEE Transactions on Industrial Electronics, vol.43,pp.519-532,1996.
5. Kennedy and R. C. Eberhart, "Particle swarm optimization," in Proc. IEEE Int. Conf. Neural Netw., Perth, Australia, 1995, vol. 4, pp. 1942–1948.

6. Y. Shi and R. C. Eberhart, “A modified particle swarm optimizer,” in Proc. IEEE Int. Conf. Evolutionary Computation, pp. 69–73, 1998.
7. Shi, X. H., et al. "An improved GA and a novel PSO-GA-based hybrid algorithm." *Information Processing Letters* 93.5 (2005): 255-261.
8. Qian, Bin, et al. "An effective hybrid DE-based algorithm for multi-objective flow shop scheduling with limited buffers." *Computers & Operations Research* 36.1 (2009): 209-233.
9. Nemati, Shahla, et al. "A novel ACO–GA hybrid algorithm for feature selection in protein function prediction." *Expert systems with applications*36.10 (2009): 12086-12094.
10. Lin, Wen-Yi. "A GA–DE hybrid evolutionary algorithm for path synthesis of four-bar linkage." *Mechanism and Machine Theory* 45.8 (2010): 1096-1107.
11. Kim, D. H., Ajith Abraham, and K. Hirota. "Hybrid genetic: Particle swarm optimization algorithm." *Hybrid Evolutionary Algorithms*. Springer Berlin Heidelberg, 2007. 147-170.
12. García-Nieto, José, Enrique Alba, and Javier Apolloni. "Noiseless functions black-box optimization: evaluation of a hybrid particle swarm with differential operators." *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*. ACM, 2009.
13. N. Hansen, A. Auger, S. Finck and R. Ros. Real-parameter black-box optimization benchmarking 2009: Experimental setup. Technical Report RR-6828, INRIA, 2009.
14. N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2010: Experimental setup. Technical Report RR-7215, INRIA, 2010
15. N. Hansen, S. Finck, R. Ros and A. Auger, “Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions,”INRIA Technical Report RR-6829, 2009.
16. S. Finck, N. Hansen, R. Ros, and A. Auger. “Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions”, Technical Report 2009/20, Research Center PPE, 2009. Updated February 2010.
17. Jingqiao Zhang and Arthur C Sanderson. Jade: adaptive differential evolution with optional external archive. Evolutionary Computation, IEEE Transactions on, 13(5):945–958, 2009.
18. A. Auger and N. Hansen. A Restart CMA Evolution Strategy With Increasing Population Size. In IEEE Congress on Evolutionary Computation, pages 1769–1776. IEEE Press, 2005.
19. Bajer, Lukáš, Zbyněk Pitra, and Martin Holeňa. "Benchmarking gaussian processes and random forests surrogate models on the BBOB noiseless testbed." *Proceedings of the Companion Publication of the 2015 on Genetic and Evolutionary Computation Conference*. ACM, 2015.
20. Holtschulte, Neal J., and Melanie Moses. "Benchmarking cellular genetic algorithms on the BBOB noiseless testbed." *Proceedings of the 15th annual conference companion on Genetic and evolutionary computation*. ACM, 2013.
21. Narayan, V., Faiz, M., Mall, P. K., & Srivastava, S. (2023). A Comprehensive Review of Various Approach for Medical Image Segmentation and Disease Prediction. *Wireless Personal Communications*, *132*(3), 1819-1848.Storn, Rainer, and Kenneth Price. *Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces*. Vol. 3. Berkeley: ICSI, 1995.
22. Dorigo, Marco, Mauro Birattari, and Thomas Stützle. "Ant colony optimization." *Computational Intelligence Magazine, IEEE* 1.4 (2006): 28-39.
23. Faiz, M., & Daniel, A. K. (2023). A hybrid WSN based two-stage model for data collection and forecasting water consumption in metropolitan areas. *International Journal of Nanotechnology*, *20*(5-10), 851-879.
24. Channi, H. K., Sandhu, R., Faiz, M., & Islam, S. M. (2023, August). Multi-Criteria Decision-Making Approach for Laptop Selection: A Case Study. In *2023 3rd Asian Conference on Innovation in Technology (ASIANCON)* (pp. 1-5). IEEE.
25. Faiz, M., Fatima, N., & Sandhu, R. (2023). A Vaccine Slot Tracker Model Using Fuzzy Logic for Providing Quality of Service. *Multimodal Biometric and Machine Learning Technologies: Applications for Computer Vision*, 31-52.
26. Sandhu, R., Singh, A., Faiz, M., Kaur, H., & Thukral, S. (2023). Enhanced Text Mining Approach for Better Ranking System of Customer Reviews. *Multimodal Biometric and Machine Learning Technologies: Applications for Computer Vision*, 53-69.
27. Sandhu, R., Bhasin, C., Faiz, M., & Islam, S. M. (2023, August). Managing E-Reviews: A Performance Enhancement Technique Using Deep Learning. In *2023 Second International Conference On Smart Technologies For Smart Nation (SmartTechCon)* (pp. 662-666). IEEE.
28. Mall, P. K., Narayan, V., Pramanik, S., Srivastava, S., Faiz, M., Sriramulu, S., & Kumar, M. N. (2023). FuzzyNet-Based Modelling Smart Traffic System in Smart Cities Using Deep Learning Models. In *Handbook of Research on Data-Driven Mathematical Modeling in Smart Cities* (pp. 76-95). IGI Global.
29. Narayan, V., Babu, S. Z. D., Ghonge, M. M., Mall, P. K., Sharma, S., Srivastava, S., ... & Tyagi, L. K. (2023). 7 Extracting business methodology: using artificial intelligence-based method. *Semantic Intelligent Computing and Applications*, *16*, 123.
30. Chaturvedi, P., Daniel, A. K., & Narayan, V. (2021). Coverage Prediction for Target Coverage in WSN Using Machine Learning Approaches.
31. Chaturvedi, P., Daniel, A. K., & Narayan, V. A Novel Heuristic for Maximizing Lifetime of Target Coverage in Wireless Sensor Networks. In *Advanced Wireless Communication and Sensor Networks* (pp. 227-242). Chapman and Hall/CRC.
32. Saxena, A., Chauhan, R., Chauhan, D., Sharma, S., Sharma, D., & Narayan, V. (2022). Comparative Analysis Of AI Regression And Classification Models For Predicting House Damages In Nepal: Proposed Architectures And Techniques. *Journal of Pharmaceutical Negative Results*, 6203-6215.