2023 5th International Conference on Energy, Power and Environment: Towards Flexible Green Energy Technologies (ICEPE) | 979-8-3503-1312-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICEPE57949.2023.10201630

Adaptive Fuzzy Campus Placement based Optimization Algorithm

Ankeshwarapu Sunil, Student Member, IEEE , Vijay Saieesh ATS, Student Member, IEEE , and Venkaiah Chintham, Senior Member, IEEE
Department of Electrical Engineering, NIT Warangal, Hanumakonda, Telangana, India
ankeshwarapu.sunil@ieee.org, vijaystudius@gmail.com, ch.venkaiah@ieee.org

***Abstract*—A novel Adaptive Fuzzy Campus Placement Optimization Algorithm (AFCPOA) is developed for solving unconstrained optimization problems. The proposed optimization algorithm is based on the concept of campus placement procedure adopted for offering a job to a student by an employer visiting campus for hiring students seeking employment. Fuzzy models are considered to depict written test and interview process. The performance of the proposed algorithm was tested on 10 benchmark optimization test functions and compared with other existing algorithms. Subsequently, the proposed algorithm is applied on IEEE 33 bus radial distribution system for optimal placement and sizing of Distributed Generators (DGs) to mitigate active power losses and voltage deviation. It is observed from the results that the proposed algorithm is more effective in comparison with existing algorithms.**

***Index Terms*—Adaptive Fuzzy Campus Placement based Optimization Algorithm (AFCPOA), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Teaching Learning based Optimization (TLBO), Jaya Algorithm (JA).**

I. INTRODUCTION

Optimization is a process of determining the solution for which the objective function has the best value. One obvious procedure is to sample the space and perform trial and error search [1]. Traditional optimization methods such as sequential search methods are slow and gradient-based methods fail when the objective function is non-differentiable or dis-continuous. Meta-heuristic techniques bypass such issues and promise to solve almost any optimization problem. Broadly, Meta-heuristic techniques are classified into 2 types – Single-based meta-heuristics and Population-based Meta-heuristics. Population based Meta-heuristics are classified into 4 types – evolutionary, physics-based, swarm intelligence-based and human-based [2].

Algorithms inspired by basic biological evolution of nature are called Evolutionary Algorithms. The basic skeleton consists of Selection, Recombination and mutation [2]. Genetic Algorithm (GA) [3] mimicks the species evolution, which is one of the foremost works in the field. Based on the theory proposed by Darwin, “Survival of the fittest”, the individuals with better fitness are considered to have higher probability of reproducing using operators like Mutation and Crossover. The Differential Evolution (DE) Algorithm [4] is a population based optimization algorithm for solving the non linear optimization problem. Several other evolutionary algorithms are listed in [2]. Algorithms inspired from established physical laws of nature are called Physics- based Algorithms. Some such algorithms include the Central Force Optimization algorithm, Gravity Search algorithm, etc. Algorithms inspired by collective behaviour of non-human creatures are called Swarm Intelligence Algorithms. Particle Swarm Optimization [5] is one of the famous Swarm Intelligence based algorithm based on the collective behaviour of a flock of birds migrating from one place to another. Each bird (particle) is influenced by the experience of other birds along with its own prior experience. Two parameters, velocity and position of the particle drive the search while encompassing the philosophy. Bat Algorithm [6] is based on echolocation ability of bats, where the search is driven by control parameters which vary depending on the distance between the bat and the prey. The Ant- Colony Optimization [7] was proposed based on the method of communication between ants in search of food source. The level of pheromone deposit drove the algorithm towards optimum. The Artificial Bee Colony Optimization [8] was based on how honey bees explore and exploit food sources. Similar to Ant Colony Optimization, the search was driven by Waggle Dance Communication.

Algorithms inspired by Human actions are called Human-based Algorithms [2]. One of the famous algorithms is the TLBO Algorithm [9] in which, the search for better solution is driven by knowledge transfer from teacher (the best student) to other students. The Jaya Algorithm [10] is proposed, as the name suggests on victory where every particle tries to move closer to the “Winner” and away from “Loser”. Apart from the algorithms listed above, numerous other algorithms exist in the literature. Authors in [11] has provided a comprehensive review of evolutionary large-scale multi objective optimization algorithms that are currently in use, along with comparisons of their performance on various benchmarks.

Recently, works combining machine learning, fuzzy logic and evolutionary optimization have been published. Effective distribution of COVID-19 vaccines before-hand is a part optimization problem which has been solved by an evolutionary algorithm in [12]. Authors in [13] has proposed an improved version of Ant Colony Optimization for SCM applications. Similarly, authors in [14] has modified the Ant Colony Optimization algorithms for electric vehicle re-routing application.

Such instances indicate that meta-heuristic techniques are useful, particularly in solving real-time optimization problems which cannot be solved by hard-computing techniques [15]. In the field of electrical engineering, many optimization problems such as - economical load dispatch, optimal placement of DG,

979-8-3503-1312-3/23/$31.00 ©2023 IEEE

optimal power flow etc. [16] are solved on regular basis. In particular, the optimal DG placement problem has been solved by authors in [17], [18].

From the literature survey, it is understood that many processes ranging from evolution of micro-organisms to process by human-actions can be viewed as an optimization strategy. To the best knowledge of the authors of this paper, optimization algorithm inspired by Campus Placement has not yet been proposed. This paper proposes a novel evolutionary algorithm based on the process of campus placement. The proposed algorithm was tested on 10 standard benchmark test functions for basic validation and applied on IEEE 33 bus radial distribution system for optimal placement and sizing of DGs to mitigate active power losses and voltage deviations.

This paper is divided into five sections: The idea for the innovative campus placement procedure which describes the philosophy of the proposed algorithm is explained in Section II. Section III explains individual steps of the proposed algorithm. The validation results of the algorithm is presented in Section IV. Conclusions and suggestions for further research are presented in Section V.

II. Inspiration - Campus Placement Process

A significant share of employment in a country like India is taken up by Multi-National Companies (MNCs) and Startups that visit campuses and recruit undergraduate and graduate students who are in their final year of study. In fact the reputation of the university is skewed towards the list of companies that visit and the salary package they offer to students. Over 98% of students want to be employed at the end of their study. They 'compete' among themselves to obtain the job offer. It is natural, that companies want the 'best' minds to work for them. The skills expected from candidates are continuously evolving with shifts in the job market paradigm. However, the philosophy behind the recruitment process remains the same.

A number of candidates apply for a job vacancy with a hope of obtaining it. Usually, a small percentage of candidates does not satisfy the technical requirements of the job. The company does not wish to waste time in training them and hence they are rejected. The remaining candidates are taken to a written test assessment which generally consists of an Aptitude test, Programming Test, Logical Reasoning etc. Solving problems is a common aspect of working for a company either looking for solutions in R&D or addressing customer problems. This requires understanding of the problem and providing quick innovative solutions. Aptitude tests consist of a series of problems to be solved in a limited amount of time. A candidate with good problem solving skills will be able to solve most of the problems correctly within the stipulated time. In the present, there is a huge demand for development and testing engineers in IT service sector. Such jobs require the skill to implement a solution digitally. The candidates are hence expected to have programming knowledge. There are certain skills the candidates have to posses which cannot be tested objectively. They have to be assessed in person. Service based companies generally interview all candidates who have completed the written component for communication skills, soft skills, etc. MNCs have offices over the globe and this necessitates the candidate to communicate effectively, which includes language and other soft skills. Also, during R&D interviews, apart from academic scores of the candidates, their research output and practical understanding of core concepts are assessed along with communication skills. The overall quality (calibre) of the candidate is generally a weighted sum of written test assessment and interview scores. However, this score is merely an entry level measurement. The selected candidates are trained (probation) for generally 6 months to 12 months so that they are perfectly fit for employment.

III. Proposed Optimization Algorithm

The campus placement process from another vantage point is optimization.
"A swarm of particles (cohort of students) through iterations of evolution (repeated training) reach the global optima (knowledge level for project deployment)"
The general placement process is slightly modified to fit into the schema of a meta-heuristic algorithm.

Step 1 - Initialisation
Initialize a cohort of *np* candidates within search space with *nv* components each. *np* is the size of initial population and *nv* is the dimension of search space.

Step 2 - Evaluation
This step is analogous to function evaluation in other algorithms. As explained in the previous section, the skills that the candidate possesses determine his/her chances of placement. The *fitness* of a candidate is very abstract and more subjective measures are defined that are directly dependant on *fitness* to make the algorithm more realistic. There are 2 assessments defined - Written test (assessment) and Interview. The written test consists of sub-components - technical knowledge (denoted by CGPA (Cumulative Grade Point Average) hereafter), aptitude and programming skills. The interview consists of sub-components - research output, projects and communication skills. The scores of individual sub-components have direct relation to CGPA with some randomness to retain uncertainty in this relation.

Written Test - The written test aims to assess objective skills such as academic knowledge, aptitude and programming skill with the following sub-components respectively.

CGPA - The academic knowledge gained by the candidate (though not accurately) is quantified by CGPA. The fitness value of candidate cannot be taken as a direct measure of CGPA. It should be mapped into a range of 6.5 to 10. Eq.(1) does this mapping. The mapping is relative, meaning, the least *fitness* is mapped to minimum cgpa and the highest *fitness* is mapped to maximum cgpa. The limits are prescribed from real-life experiences.

$$\begin{aligned} cgpa = &(candi\_fitness - mini\_candi\_fitness) \\ &/(maxi\_candi\_fitness - mini\_candi\_fitness) \\ &*(max\_cgpa - min\_cgpa) + min\_cgpa \end{aligned} \tag{1}$$

Aptitude Test - The Aptitude test assesses the problem solving speed of the candidate. In reality, mathematical problems from a wide range of topics including but not limited to Number

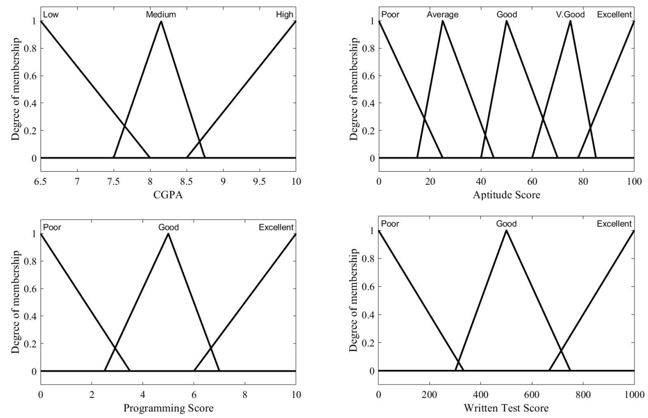

Fig. 1: Membership functions of written test

Systems, Profit and Loss, elementary mathematics, Permutation & Combination etc. are used. Generally the stipulated time for this section is 3/4th of the actual time required to solve all questions. Even though the candidates probably know standard formulae to solve most of the questions, the ability to recall them and arrive at the right answer is tested. The algorithm considers a score range of 0 to maximum aptitude score, so that candidates can be distinguished from one another quantitatively. The Aptitude score is calculated by Eq.(2):

$$aptitude = max\_aptitude\_score * cgpa/max\_cgpa \tag{2}$$

Programming - Unlike aptitude, the programming skill of the candidate cannot be deterministically related to CGPA. The candidate may give an extraordinary solution to the problem. But, the ability to code it digitally is not certain. But, to relate with the fitness of the candidate and to simultaneously incorporate randomness, Eq.(3) is defined.

$$prog\_score = rand * max\_prog\_score * cgpa/max\_cgpa \tag{3}$$

While obtaining the final score of the candidate in written test, a human uncertainty factor which cannot be quantified directly exists. Hence, an FIS (Fuzzy Inference System) is employed to mitigate this.
A fuzzy inference engine with rule base is designed based on common sense which yields the written test score based on scores obtained in the individual components. The membership functions of all fuzzy variables are triangular for simplicity, as shown in Figure 1. The fuzzy rules designed for written test part is tabulated in Table I. The output of fuzzy engine is denoted as *written_score*.

Interview - The written test assesses objective skills. But, apart from those, each candidate is expected to have a bunch of subjective skills which is difficult to assess without personal interaction. Different companies have different levels of interviews, Technical rounds, HR etc. However, the proposed algorithm is designed to assess the research output, projects and communication skills of candidates.
Research Output - Research Output of a candidate aims to assess the curiosity of a candidate to know latest developments in the field and contribute to it. This is quantified by the technical conferences he/she attends, the quality of publications, etc. Eq.(4) quantifies the research output of the candidate. Here, a linear relation between CGPA and research output is presented. Logically, a person with low CGPA cannot be curious about research. However, it is tedious to concentrate on research and academics simultaneously efficiently. Hence, a research concentration factor is introduced in Eq.(4).

$$\begin{aligned} research\_score = research\_conc\_factor* \\ max\_research\_score * cgpa/max\_cgpa \end{aligned} \tag{4}$$

Projects - The CGPA of the candidate reflects mostly the theoretical knowledge gained. Term projects are evaluated to test hands-on experience gained. Novel projects require out of the box thinking. But to carry out novel projects, the understanding of concepts is necessary. Similar to the trade off between CGPA and research output, a project concentration factor is introduced for evaluating project score. This score is calculated using Eq.(5)

$$\begin{aligned} proj\_score = proj\_conc\_factor * max\_proj\_score \\ *cgpa/max\_cgpa \end{aligned} \tag{5}$$

Communication Skills - Companies have offices across the globe and employees are often required to communicate with colleagues overseas. Even, inland, the skill to convey thoughts accurately is a necessity. This skill is assessed in the way the candidate answers, presents himself, conveys his/her thoughts etc. to the interviewer. Similar to programming skill, communication skill also has a random factor associated with it as shown in Eq.(6):

$$\begin{aligned} comm\_skill\_score = rand* \\ max\_comm\_skill\_score * cgpa/max(cgpa) \end{aligned} \tag{6}$$

The final interview score of candidate denoted as *int_score* is calculated considering the human uncertainty similar to written test score. A fuzzy inference engine is designed based on common logic taking the above sub-components as input. Triangular membership functions are employed for simplicity with an appropriate range shown in Figure 2. The fuzzy rules designed for interview part are tabulated in Table II.

TABLE I: Fuzzy rules designed for Written test

Rules	Inputs			Output
	CGPA	Aptitude	Coding	Written Test score
Rule 1	Low	Poor	Poor	Poor
Rule 2	Low	Poor	Good	Poor
...	...	...	...	...
Rule 9	Low	Good	Excellent	Excellent
...	...	...	...	...
Rule 25	Medium	Very Good	Poor	Good
...	...	...	...	...
Rule 35	High	Average	Good	Good
...	...	...	...	...
Rule 45	High	Excellent	Excellent	Excellent

The final score of the candidate is a weighted sum of written test and interview scores with separate weighting factors. This

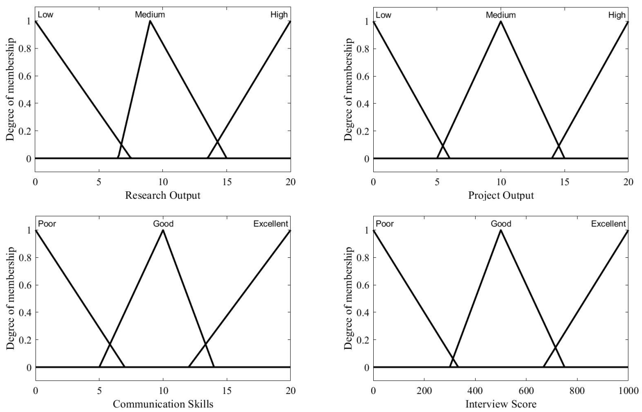

Fig. 2: Membership functions of Interview

TABLE II: Fuzzy rule table for Interview

Rules	Inputs			Output
	Research	Project	English	Interview score
Rule 1	Low	Low	Poor	Poor
Rule 2	Low	Low	Good	Poor
…	…	…	…	…
Rule 13	Medium	Medium	Poor	Good
…	…	…	…	…
Rule 20	High	Low	Good	Excellent
...	…	…	…	…
Rule 27	High	High	Excellent	Excellent

score is denoted as *calibre* and is defined as in Eq.(7)

$$calibre\_score = wt\_weightage * wt\_score + int\_weightage * int\_score \tag{7}$$

Step 3 - Probation

The knowledge level of each candidate is indexed by *calibre* from Eq.(7). From the cohort, ut (untrainable) % of candidates having lowest calibre scores are eliminated from the process. Such candidates are termed "untrainable". The remaining candidates, called "Trainable" undergo training. Training is to update the knowledge level from current level, similar to particle movement to a better fitness position. It it to be noted that the calibre score is relative and it so happens that, even a candidate with highest calibre score (denoted as the elite), has low knowledge level. The Probation phase is divided into 2 sub-phases. Firstly, the *elite* candidate is given a number of chances to improve until there is evidence of improvement. The *elite candidate* is trained by Eq.(8). Numerically, increment in position of *elite candidate* in every attempt follows a saturated increasing trend with minimum and maximum bounds and is constrained to be much lower than the range of search space. This ensures that the increase is not too large to jump the global optima while at the same time accelerate the movement with attempts while not allowing the coordinates to saturate. Improvement is measured in terms of *fitness* of *elitenew* after which the second phase of probation is executed.

TABLE III: Statistical parameters obtained by novel AFCPOA on 10 unconstraint benchmark test functions (SD: standard deviation)

F.No	Function Name	Optimum	Best	Worst	Mean	SD
F1	Ackley	0	0	0	0	00E+00
F2	Branin	0.398	0.397887	0.399789	0.397952	0.000347
F3	Colville	0	0	0	0	00E+00
F4	Fletcher Powell2	0	0	0	0	00E+00
F5	Foxholes	0.998	0.998005	1.0868	1.014661	0.029669
F6	Langermann5	-1.08	-1.08	-1.08	-1.08	0.000
F7	Michalewicz2	-1.8013	-1.8013	-1.8013	-1.8013	00E+00
F8	Perm	0	0	0	0	00E+00
F9	Schwefel1.2	0	0	0	0	00E+00
F10	Zakharov	0	0	0	0	00E+00

$$perturb\_elite\_candi = elite\_candi + sigmoid(attempt) * (-ulim + 2 * ulim * rand)/100 \tag{8}$$

During the second phase of probation, the remaining trainable candidates are trained towards *perturb_elite_candi*. This is performed following Eq.(9). Philosophically, this equation model peer-to-peer interaction or knowledge transfer which is proved to be very effective.

$$perturb\_candi = candi + (-1 + 2 * rand) * (elite\_candi - abs(candi)) \tag{9}$$

Step 2 and Step 3 are repeated until the algorithm converges. The foundational philosophy of the algorithm is to uplift candidates knowledge level to optimum. This means that, at convergence, the knowledge level of candidates is equal and is in fact optimum.

IV. Results and Discussion

The proposed AFCPOA's performance was tested using MATLAB -R2022b software on a computer with an Intel (R) Core (TM) i7-3770 CPU running at 3.40GHz and 8GB of RAM on 10 benchmark test functions that were available in the literature for optimization. The benchmark functions have different characteristics like uni model/multi model, separable/ nonseparable, regular/ non regular...etc. The number of design variables and their ranges is different for each problem. The proposed AFCPOA's results have been compared with JA, ABC, TLBO, DE, PSO, and GA. The proposed AFCPOA is run 30 times for each benchmark test function, same as other algorithms, and the mean and standard deviation values are compared with other algorithms for the same number of runs. The best values of the proposed AFCPOA approach that has been suggested are given in Table III for 10 benchmark functions. Table IV presents a comparison of statistical parameters of proposed algorithm among some of the existing algorithms. The proposed technique was executed for 30 independent trials with a maximum of 240000 function evaluations to maintain consistency in comparisons. Comparing the best, worst, mean, and standard deviation of the converged optima produced by the proposed method, it was found that the novel AFCPOA outperforms other algorithms on the functions in terms of convergence of the global optima, robustness to

```
Initialize population;
Determine fitness;
Resume Filtering;
while convergence do
    for candidate do
        calculate cgpa;
        calculate aptitude_score;
        calculate programming_score;
        written test FIS;
        obtain wt_score;
        calculate research_score;
        calculate project_score;
        calculate comm_skill_score;
        interview FIS;
        obtain int_score;
        calculate calibre_score;
    end
    choose elite_candidate;
    choose worst_candidate;
    choose gbest,gbest_fitness;
    set better=0;
    while better==0 && attempt ≤ chance do
        calculate perturb_elite_candidate;
        calculate perturb_fitness;
        if perturb_fitness ≤ gbest_fitness then
            update elite_candidate;
            set better=1;
        else
            set attempt=attempt+1;
        end
    end
    Discard untrainable_candidates;
    for candidate do
        calculate perturb_candidate;
        calculate perturb_fitness;
        if perturb_fitness ≤ candidate_fitness then
            update candidate;
            update candidate_fitness;
        else
        end
        if candidate_fitness ≤ gbest_fitness then
            update gbest;
            update gbest_fitness;
        else
        end
    end
end
```

Algorithm 1: Pseudo code of AFCPOA

TABLE IV: Comparison of the Statistical Parameters of AFCPOA with other algorithms on Benchmark Test Functions

F. No	Statistic	GA [3]	PSO [5]	DE [4]	ABC [8]	TLBO [9]	JA [10]	AFCPOA (Proposed)
F1	M	14.67178	0.164622	0	0	0	0	0
	SD	0.178141	0.493867	0	0	0	0	0
	Best	-	-	-	-	-	0	0
F2	M	0.397887	0.397887	0.397887	0.397887	0.397887	0.397887	0.397952
	SD	0	0	0	0	0	0	0.000347
	Best	-	-	-	-	-	0.398	0.397887
F3	M	0.014938	0	0.040912	0.092967	0	0	0
	SD	0.007364	0	0	0.081979	0.066277	0	0
	Best	-	-	-	-	-	0	0
F4	M	0	0	0	0	-	-	0
	SD	0	0	0	0	-	-	0
	Best	-	-	-	-	-	0	0
F5	M	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	1.014661
	SD	0	0	0	0	0	0	0.029669
	Best	-	-	-	-	-	0.998	0.998005
F6	M	-1.08	-0.67927	-1.08	-1.08	-1.08	-1.08	-1.084
	SD	0	0.274621	0	0	0	0	0
	Best	-	-	-	-	-	-1.08	-1.08
F7	M	-1.8013	-1.57287	-1.80130	-1.80130	-1.80130	-1.80130	-1.8013
	SD	0	0.11986	0	0	0	0	0
	Best	-	-	-	-	-	-1.8013	-1.8013
F8	M	5.870093	3	3	3	3	3	0
	SD	1.071727	0	0	0	0	0	0
	Best	-	-	-	-	-	0	0
F9	M	7.40E+03	0	0	0	0	0	0
	SD	1.14E+03	0	0	0	0	0	0
	Best	-	-	-	-	-	0	0
F10	M	0.013355	0	0	0.0002476	0	0	0
	SD	0.004532	0	0	0.000183	0	0	0
	Best	-	-	-	-	-	0	0

TABLE V: Comparison of AFCPOA with other algorithms for optimal placemnet and sizing of DGs in IEEE 33 bus distribution system

Algorithm	GA	PSO	DE	ABC	TLBO	JA	AFCPOA
DG_{loc}	19 20 18	13 24 30	14 25 30	6 15 25	13 24 30	31 24 16	24 14 30
DG_{size} (kW)	655 520 532	790 1070 1010	733.9 733.5 1032.6	1756.9 575.7 782.6	798.8 1090.28 1053.42	850 1000 920	1102 757 1075
P_{loss}(kW)	150.5	75.74	72.90	79.2526	72.77	61.60	**57.38**
% P^{red}_{loss}	35.5	64.1	65.44	61.134	65.51	70.8	**72.8**
V_{min} (p.u)	0.9364	0.9641	0.9658	0.9524	0.9684	0.9682	**0.9686**

variation in the initial population, and convergence rate. The algorithm has a unique characteristic that causes the average quality of candidates to gradually grow while the size of the swarm is significantly reduced at each iteration because the worst-performing candidates are systematically eliminated. Fortunately, the improvement in average quality surpasses the reduction in population size, suggesting that the algorithm still tries to attain the global optima with fewer function evaluations. This significantly reduces the computational effort.

A. Case Study: Application of proposed AFCPOA for optimal placement and sizing of DGs into distribution system

The optimal placement and sizing of multiple DGs into the IEEE 33 bus radial distribution system is used as a case study to show how well the proposed method performs when used to address a real-world electrical engineering problem. The backward forward sweep concept for distribution load flow as employed in [19] is utilized for the process of problem-solving by applying the proposed method. The proposed AFCPOA application for the optimal placement and sizing problem is drawn from [18] to minimize the total system losses and reduce deviations in voltage in the distribution network.

By integrating DGs into the IEEE 33 bus RDS, there will be reduction of overall active power losses and voltage deviation. The findings by including the optimal DG placement and sizes in the IEEE 33 bus radial distribution system, are listed in Table V and the performance of the proposed AFCPOA is compared with other existing methods. It is observed from backward forward sweep concept based distribution load flow study on IEEE 33 bus radial distribution system without integration of DGs, an active power loss of 210.99 kW, reactive power loss of 143.03 kVAR and a minimum bus voltage of 0.9192 p.u. By integrating 3 DGs at optimal locations and sizes (Bus24:1102kW; Bus14: 757kW and Bus30:1075kW) utilizing AFCPOA, it is observed that there is a reduction of active power loss from 210.99 kW to 57.38 kW and improvement of minimum bus voltage from 0.9192 p.u to 0.9686 p.u. For optimal placement and sizing of multiple DGs in the IEEE 33 bus radial distribution system, the convergence plot of the proposed AFCPOA and other methods is shown in Figure

V. Conclusion and Future Works

A novel meta-heuristic algorithm i.e. AFCPOA was proposed drawing inspiration from the campus placement process. Each step of the proposed algorithm has been realistically designed considering randomness and logic in relation to

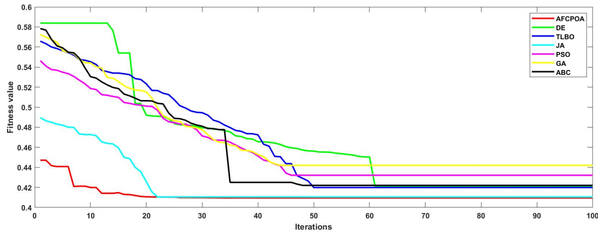

Fig. 3: Convergence plots of various optimization algorithms for optimal placement and sizing of DGs

various skills a candidate possesses. It would be unrealistic to determine the written test and interview score as they are affected by highly subjective factors which cannot be quantified. Hence, a fuzzy model which tends to inculcate this subjective nature in candidate performance was employed. The algorithm was validated on a bunch of 10 benchmark test functions with different combinations of properties such as Uni-modality, Multi-modality and Separability and Inseparability. The proposed AFCPOA was successfully applied to a real-world electrical engineering optimization problem, specifically the optimal location and size for multiple DGs in a distribution system for planning and operational studies. The performance of the proposed novel AFCPOA was compared with that of other existing algorithms and it was observed that the proposed AFCPOA performs better than others on almost all functions. Through the algorithm was designed to discard significant percentage of badly performing candidates at every iteration, the number of function evaluations was much lower compared to that of other algorithms, without compromising the accuracy of global solution that converged, which reduced computational burden significantly. Future research may examine the multi objective optimal placement of DG while taking into account various DG penetration levels and load needs. Additionally, using uncertainty modelling, the impact of the intermittent nature of renewable DG might be addressed.

References

[1] O. Bozorg-Haddad, M. Solgi, and H. A. Loáiciga, *Meta-heuristic and evolutionary algorithms for engineering optimization*. John Wiley & Sons, 2017.

[2] Y. Meraihi, A. B. Gabis, S. Mirjalili, and A. Ramdane-Cherif, “Grasshopper optimization algorithm: theory, variants, and applications,” *IEEE Access*, vol. 9, pp. 50001–50024, 2021.

[3] J. H. Holland, “Genetic algorithms,” *Scientific american*, vol. 267, no. 1, pp. 66–73, 1992.

[4] K. Fleetwood, “An introduction to differential evolution,” in *Proceedings of Mathematics and Statistics of Complex Systems (MASCOS) One Day Symposium, 26th November, Brisbane, Australia*, pp. 785–791, 2004.

[5] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95-international conference on neural networks*, vol. 4, pp. 1942–1948, IEEE, 1995.

[6] X.-S. Yang and A. H. Gandomi, “Bat algorithm: a novel approach for global engineering optimization,” *Engineering computations*, 2012.

[7] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” *IEEE computational intelligence magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[8] D. Karaboga, “Artificial bee colony algorithm,” *scholarpedia*, vol. 5, no. 3, p. 6915, 2010.

[9] R. V. Rao, V. J. Savsani, and D. Vakharia, “Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems,” *Computer-aided design*, vol. 43, no. 3, pp. 303–315, 2011.

[10] R. Rao, “Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems,” *International Journal of Industrial Engineering Computations*, vol. 7, no. 1, pp. 19–34, 2016.

[11] S. Liu, Q. Lin, K.-C. Wong, Q. Li, and K. C. Tan, “Evolutionary large-scale multiobjective optimization: Benchmarks and algorithms,” *IEEE Transactions on Evolutionary Computation*, 2021.

[12] Y.-J. Zheng, X. Chen, Q. Song, J. Yang, and L. Wang, “Evolutionary optimization of covid-19 vaccine distribution with evolutionary demands,” *IEEE Transactions on Evolutionary Computation*, 2022.

[13] X. Zhang, Z.-H. Zhan, W. Fang, P. Qian, and J. Zhang, “Multipopulation ant colony system with knowledge-based local searches for multiobjective supply chain configuration,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 3, pp. 512–526, 2022.

[14] Y.-H. Jia, Y. Mei, and M. Zhang, “Confidence-based ant colony optimization for capacitated electric vehicle routing problem with comparison of different encoding schemes,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 6, pp. 1394–1408, 2022.

[15] M. Nutakki and S. Mandava, “Review on optimization techniques and role of artificial intelligence in home energy management systems,” *Engineering Applications of Artificial Intelligence*, vol. 119, p. 105721, 2023.

[16] J. Qian, P. Wang, C. Pu, X. Peng, and G. Chen, “Application of effective gravitational search algorithm with constraint priority and expert experience in optimal allocation problems of distribution network,” *Engineering Applications of Artificial Intelligence*, vol. 117, p. 105533, 2023.

[17] E. A. Al-Ammar, K. Farzana, A. Waqar, M. Aamir, A. U. Haq, M. Zahid, M. Batool, *et al.*, “Abc algorithm based optimal sizing and placement of dgs in distribution networks considering multiple objectives,” *Ain Shams Engineering Journal*, vol. 12, no. 1, pp. 697–708, 2021.

[18] A. Sunil, S. Kongala, and C. Venkaiah, “Metaheuristic techniques based optimal placement and sizing of multiple distributed generations in radial distribution system,” in *2021 13th IEEE PES Asia Pacific Power & Energy Engineering Conference (APPEEC)*, pp. 1–6, IEEE, 2021.

[19] A. Sunil and V. Chintham, “Investigations on distribution load flow methods in a radial distribution network,” in *2022 IEEE Students Conference on Engineering and Systems (SCES)*, pp. 01–06, 2022.