Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 1): pp. 7388-7402
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.01.764

Voltage Stability Analysis In Distribution System Predetermination

Parvesh Saini Department of Electrical Engineering, Graphic Era Deemed to be University, Dehradun, Uttarakhand India, 248002, parvesh.saini.eee@geu.ac.in

Ashutosh Dixit Department of Electrical Engineering, Graphic Era Deemed to be University, Dehradun, Uttarakhand India, 248002, ashutoshdixit@geu.ac.in

Sandeep Sunori Department of Electronics & Communication Engineering, Graphic Era Deemed to be University, Dehradun, Uttarakhand India, 248002, sandeepsunori@gmail.com

Abstract

Electrical power systems are experiencing severe stability issues as a result of the ongoing demand for power usage. Incorporating distributed systems with DG, Solar, Wind, and Microgrids close to consumer sites has been one of the workable solutions to reduce losses through peak shaving, overloading, and increased reliability. to investigate the impact of generator integration on the radial bus system's voltage profile and to make sure that one of the distributed generating systems mentioned above is installed in the best possible location for the best possible outcomes.

Three alternative algorithms—the Shuffled Frog Leap Algorithm (SFLA), the Artificial Bee Colony (ABC), and the Adaptive Particle Swarm Optimization—are the focus of this project (APSO). Based on the input line and bus data, these algorithms are used to analyse the week bus, the ideal quantity of active power that has to be injected, and the voltage profile of the entire bus before and after injection of active power in the radial bus system.

Artificial bee colony (ABC)

Introduction

The artificial Bee Colony (ABC) algorithm, developed by Karaboga in the year 2000, is one of the most important meta-heuristic algorithms. It reduces the difficulty of solving mathematical issues that are solely dependent on honeybee foraging behaviour. There are three main components: food or sustenance supplies, idle forager bees, and working forager bees. The majority of the foraging bees search for abundant food sources close to their colony. Additionally, the algorithm specifies significant behaviour modes that are crucial to the process. Hiring new foragers to wealthy food resources will result in favourable reviews, and desertion of poor food supplies based on unfavourable feedback from the foragers will result in self-organizing and collective intelligence.

In this forager bees look for wealthy food supplies (searching optimal solutions for a specific problem). This methodology concentrates on minimizing the objective function by converting the problem into the simplest vector. Then, the bees find a series of solution vectors in a random manner and then betters them by increasing the number of iterations by deploying the strategies of marching towards better solutions using neighbour search and neglecting the poor solutions.

Meta-heuristic

In ABC, the bees are classified into three teams: employed bees deployed at food supplies, onlooker bees looking at the work of deployed bees to decide on a nourishment supply and scout bees checking out food supply arbitrarily. Scouts and Onlookers bees are also referred to as unemployed bees. At First, scout bees detect the entire food supply positions. There-after, the amount of food supplies is utilized by employed and witness bees, and this process exhausts them. After this, the exhausted bees become scout bee and repeats the process all over again. Here, the potential solution to the issue and their excellence (fitness) is denoted by the position and quantity of supplies. the amount of deployed bees is equivalent to the amount of food supplies. Every utilized bee is dedicated only to a single food particle.

The final arrangement of ABC's algorithmic program is as follows:

1. Initializing

2. Repetition of

    - Employed Bees stage
    - Onlooker Bees stage
    - Scout Bees stage

3. Study the simplest solution attained thus far

Initializing stage

Scout bees initialize the entire vectors of the populace of food supplies ie x_m's, where m=1,2,3……SN and SN is the amount of population and management factors have been set up. Since every food supply(xm), could be a solution vector to the optimizing issue, every x_m vector retains n variables, (xmi, i=1,2…..), that are being enhanced thus on minimizing the target function.

The following expression may be utilized to initialize the functions (5):

$$xmi = li + rand(0, 1) * (ui - li)$$

Where ui and li and are the upper and lower bound of the parameter x¬m correspondingly.

$$p_m = \frac{fit_m(\vec{x_m})}{\sum_{m=1}^{SN} fit_m(\vec{x_m})}$$

Employed Bee phase

Employed bees hunt fresh food supplies (vm) with a lot of nectar among the neighbourhoods of food supplies (xm) in memory. They notice a neighboring food supply vm so assess its profit (fitness). for instance, they will verify neighbor food supply exploiting the formula:

$$\upsilon mi = xmi + \phi mi\,(xmi - x\,ki)$$

Where i could be selected parameter index in a random manner and ϕmi could be a variety among the range [-a, a] in a random manner and xk could be chosen as food supply in a random manner. When manufacturing the innovative food supply vm, its fitness is computed, and a greedy choice is utilized among xm and vm.

$$\upsilon mi = xmi + \phi mi\,(xmi - x\,ki)$$

The fitness price of the answer, fitm(xm), may be computed for step-down issues exploiting the subsequent formula: optimization issue, every xm vector retains n variables, (xmi, i=1, 2…..), which must be optimized thus on minimizing the target function.

$$fit_m(\vec{x_m}) = \begin{cases} \dfrac{1}{1 + f_m(\vec{x_m})} & \text{if } f_m(\vec{x_m}) \geq 0 \\ 1 + abs(f_m(\vec{x_m})) & \text{if } f_m(\vec{x_m}) < 0 \end{cases}$$

Where fm (xm) is the target perform price of resolution xm.

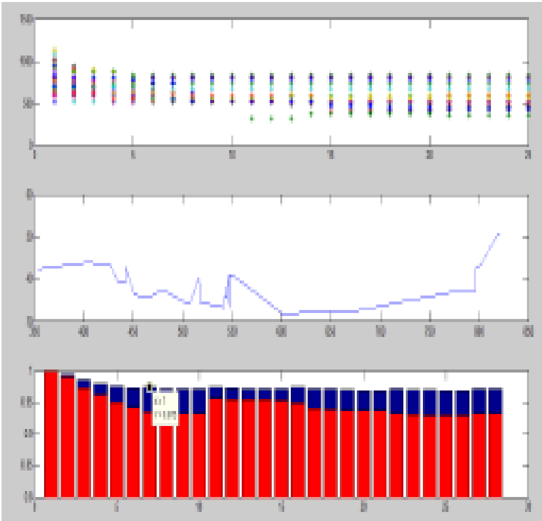
Onlooker bees' part

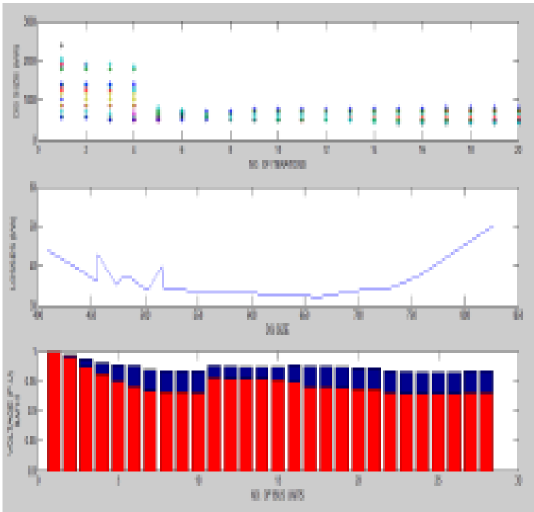
Unemployed bees comprises of an onlooker and scouts' bees. Deployed bees are sharing their own food supply data with witness bees, so they cab probabilistically select their food supplies depending on data provided to them. In ABC, associate witness bee selects a food supply depending upon the likelihood values computed by the exploitation of the fitness values offered by deployed bees. Therefore, the roulette wheel selection methodology is utilized which defines the fitness (quality) curve.

Following a food supply xm for an onlooker bee is selected by probability, a neighbourhood supply vm is set by exploitation equation, and fitness value for it is calculated. As within the employed bee's part, a greedy choice is utilized among xm and vm. Consequently, a lot of onlookers are hired to exploit richer sources which in turn creates positive feedback.

Result – Tabulation & Graph

DESCRIPTION	NUMBER OF ITERATIONS		
	50	50	50
No. of samples taken	25	25	50
Optimum location of bus	6	7	7
Determined active power to be injected(kW)	663.8	608.6	599.6
Dg losses	32.14	31.77	31.77
Measured voltage/nominal voltage (p.u) of the located week bus before active power injection	0.941	0.934	0.934
Measured voltage/nominal voltage (p.u) of the located week bus after active power injection	0.976	0.974	0.973





Shuffled frog leap algorithmic rule

Introduction

Shuffled Frog leap algorithmic rule (SFLA) may be a heuristic search algorithmic rule which was initiated by Eusuff and Lansey in 2003. The most purpose of this algorithmic rule is to determine a way to resolve difficult improvement issues with none use of ancient numerical optimization tools. In fact, this algorithmic rule is a mixture of “meme-based genetic algorithmic rule” and “Particle Swarm improvement (PSO)”. This algorithmic rule has been galvanized from the memetic development of a bunch of frogs once looking for food. In this technique, an answer to the problem is being considered within the form of a string, known as “frog” which is thought about as a control vector. The beginning populace of frogs is segregated into teams or sets known as “meme complexes”. Therefore, the variety of frogs in each and every set are equal. The SFL algorithmic rule relies on two search procedures: native search & international data exchange procedures. Supported native search, each frog improves positions to own a lot of foods (to achieve the most effective solution). In the second procedure, every subset of data will be interconnected (after a native search in subsets).

The SFL algorithmic rule sums up the advantages of the PSO and Memetic algorithms. In the SFL, the populace comprises of a group of frogs (solutions) that are classified into subsets mentioned as meme complexes. Various meme complexes are thought about as various traditions of frogs, every activity an area search within every meme complex, separate frogs hold thoughts, that can impact the thoughts of different frogs and develop through a method of memetic development. when an outlined variety of memetic development measures, concepts were passed among meme complexes in a very shuffling method. The native search and also the shuffling procedures proceed till outlined convergence standards are glad. A primary populace of P frogs is formed indiscriminately. For S-dimensional issues (S variables), a frog i is depicted as Xi=(xi1, xi2,., xiS). Afterward, the frog's ar arranged in a very downward-sloping order in line with their fitness. Subsequently, the whole populace is split into m meme complexes, each comprises n frog (i.e. P=m x n). During this method, the primary frog is going towards primary meme complex, the subsequent frog switches into following meme complex, frog m is going towards mt^{h} meme complex, and frog m+1 return to primary meme complex, and so on. At intervals of every meme complex, the frogs with the most effective fitness value are named as Xb and the worst fitness’s are known as Xw, respectively. Also, the frog with the global optimal fitness is named as Xg. Subsequently, a method just like PSO is being employed to the boost solely the frog with the worst fitness (not every frogs) in every cycle. consequently, the location of the frog with the worst fitness is altered as

Alteration in the frog position (Di) = rand (x) (Xb – Xw)

Latest position (Xw) =existing position (Xw) + Di;

Dmin ≤Di ≤Dmax

where rand() may be a variable in the range of zero to one; and D_{max} is that the most permitted amendment in every frog's position. If it produces an improved answer, it substitutes the worst frog. If Not, the computations are recurrent however concerning the worldwide best frog. During this case, if there is no progress, a brand-new answer is indiscriminately generated to interchange frogs. The computations then proceed for a selected variety of iterations consequently, the most factors of SFL are a variety of frogs P; a variety of meme complexes; Fig. 6.1 variety of production for every meme complex before shuffling; a variety of shuffling iterations; and most step size.

Solution steps for SFLN –

STEP 1: Primarily create the random populace of k frogs within the feasible area.

STEP 2: Rank Frogs Compute the fitness value of every single frog as per the issue and order them. Recording best frog position X_b in the complete populace.

STEP 3: Segregate frog to p meme complexes each having q frogs i.e. k = product (p, q). The segregation is performed that 1st frog gets allocated to initial meme complexes, likewise, frog "p" is allocated p^{th} meme complexes and (p+1) frogs allocated to the initial meme complex.

STEP 4: Identify the Memetic Development of best frog "Xb" and worst frog "Xw" and also identify the frog with global fitness as "Xg". Then worst frog can be improved as specified below

$Bi = rand (.) * (Xb - Xw)$

$New\ Xw = old\ Xw + Bi$

$-Bmax \leq Bi \geq Bmax$

Where rand (.) is in the range of zero to one.

B_{max} = Maximum allowed change.

If evolution produces a improved frog it will replace the adult frog. Otherwise, Xb replaces Xg in the above equation and the process has to be repeated.

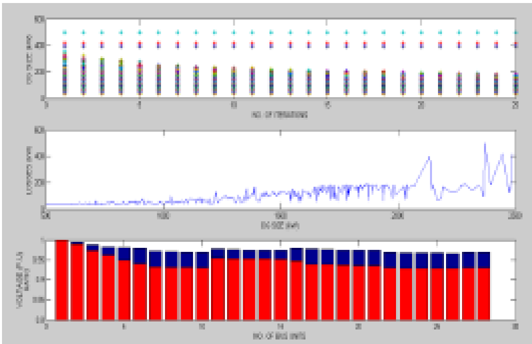
STEP 5: Local Evaluation Run the process from step 3 for a particular number of iterations.

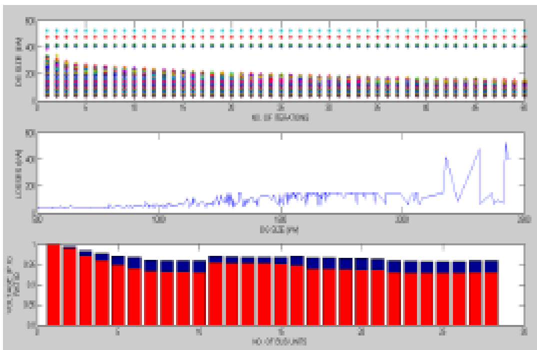
STEP 6: Perform the shuffling process by merging entire frogs in every single meme complex into one set after a demarcated number of evolutionary stages.

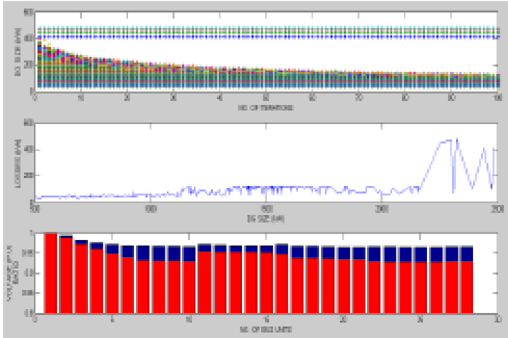
STEP 7: Verify Convergence measures, if measures are fulfilled, Terminate. If not, repeat from step 2.

As SFLA is a developing algorithm, there is no specific theory for setting the factors. We must stick to experiments. To balance the effectiveness and research capabilities, a wide range of experiments need to be conducted before finding efficient factors through errors and experiments.

DESCRIPTION	NUMBER OF ITERATIONS		
	25	**50**	**100**
Optimum location of bus	6	6	7
Determined active power to be injected(kW)	708	531	527
Dg losses	32.22	33.14	32.15
Measured voltage/nominal voltage (p.u) of the located week bus before active power injection	0.941	0.941	0.934
Measured voltage/nominal voltage (p.u) of the located week bus after active power injection	0.969	0.962	0.969







Adaptive particle swarm optimization

Introduction

The Standard Particle Swarm Optimization is a very simple algorithmic methodology. It is quite inefficient in determining the global ideals for complex issues. A version of PSO was forecasted by Kennedy and Eberhart without utilizing the speed of the earlier iteration. The speed update law can be statistically expressed as

$$Vidk+1=C1rand*(p_{best}id-S_{idk}) + C2rand*(g_{best}d-S_{idk})$$

Here, the first part, known as the “cognitive” component

The subsequent part is identified as the “social” component (global optimal)

The social component pulls the particles towards the Gbest. So, the C1and C2 are essential for the efficiency of the algorithm.

In this APSO, time-dependent acceleration coefficients are accustomed to enhancing the worldwide search within improvement stages and to inspire the particles to unite to a global ideal at the top of the search. The acceleration coefficients vary as per below expression

$$C_1=C_{min}+(C_{max}-C_{min})\, e^{(〖-(4k/G)〗^2)}$$

$$C_2=C_{min}-(C_{max}-C_{min})\, e^{(〖-(4k/G)〗^2)}$$

Here, Cmax and Cmin are taken respectively 2.5 and 0.5.

In this APSO, we can balance the social component and cognitive component, by altering the acceleration coefficients C1and C2 in the time domain.

At the first, few iterations assume that the particle i has the most excellent global position, then the particle z will also be “flying” at the speed 0, which is, it will continue till a different particle undertakes the most excellent global position. Simultaneously, some other particle is going to be “flying” to its weighted center of mass of their best position and therefore the global best position of the populace. Proposed selection for constant c1 and c2 is 2 since it makes the average masses for “social” and “cognition’’ parts to be 1. In these circumstances, the particles mathematically deal swarm to the present global best possible position until a different particle assumes the entire particles mathematically deal to the new global best position. Consequently, this can be imagined that the initial part is a procedure where the search area mathematically drops through resembling a local search algorithm.

PSO intends to display a local search w/o initial part. But, by summing the initial part, the particles tend to increase the search area, that is, they can examine the new region. So, this more probably has a global search capability by combining the first part. Both searches provide benefits for problem-solving techniques. The balances must be different among the global and local searchability for various problems. Inertia weight w performs the responsibility of balancing the global & local search. This can be fixed or a function positive by nature, non-linear or linear with respect to time.

Solution steps for APSO

STEP 1: Input data parameters for bus voltage limits, bus & Line.

STEP 2: Analyze the losses utilizing backward-forward sweep.

STEP 3: Initial array of particle populations has to be generated at random locations with dimensional velocity in the solution space. Assign the iteration k = 0.

STEP 4: For every particle, check its voltage of bus, if the voltage is within defined limits, evaluate the losses. If the voltage is not in the limits that particular particle is not feasible.

STEP 5: Compare both the individual best and every single particle objective value. If the objective value is less in comparison to Pbest, then assign this value as current Pbest, and register its appropriate position.

STEP 6: Select the particle with the minimal individual best as Pbest and assign the value of this Pbest as the current. The total best particle is the Gbest.

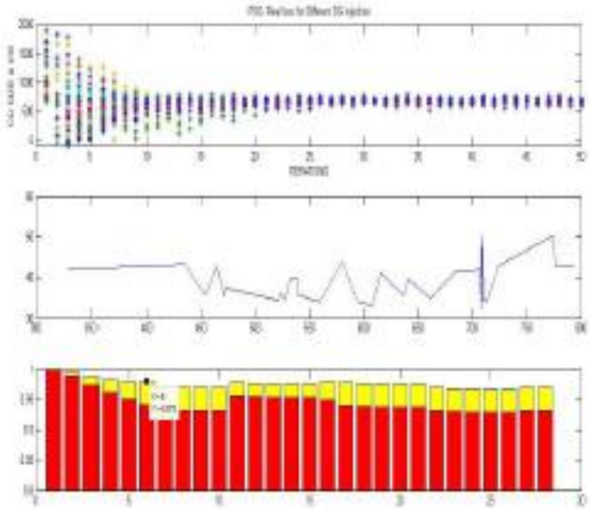
STEP 7: Updating the speed & position of the particle utilizing abovesaid equations

STEP 8: If the iteration achieves the maximal limit provided by the user, move to Step 9. Or Else, assign iteration index k = k + 1, and loop return to step 4.

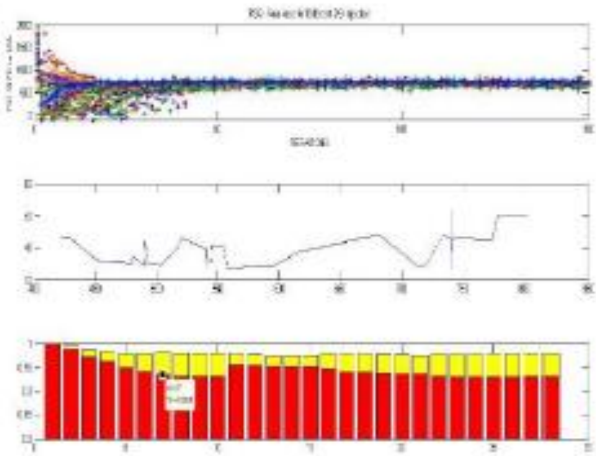
STEP 9: Print the best possible solution. The prime position comprises the DG size, optimum location of the bus, and fitness value are representing the minimal losses of power.

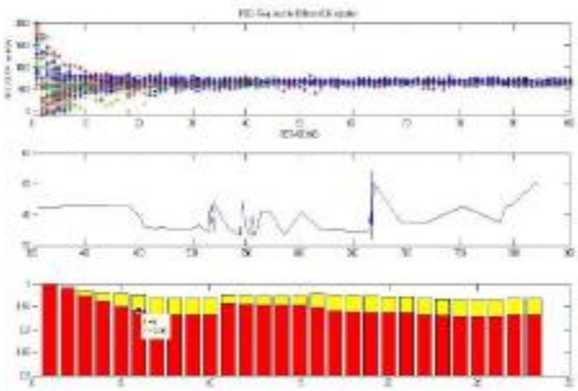
Result – Tabulation & Graph

DESCRIPTION	NUMBER OF ITERATIONS		
	25	**50**	**100**
Optimum location of bus	6	6	7
Determined active power to be injected(kW)	708	531	527
Dg losses	32.22	33.14	32.15
Measured voltage/nominal voltage (p.u) of the located week bus before active power injection	0.941	0.941	0.934
Measured voltage/nominal voltage (p.u) of the located week bus after active power injection	0.969	0.962	0.969



Comparison of APSO, ABC & SFLA outputs





DESCRIPTION	NUMBER OF ITERATIONS					
	APSO		**ABC**		**SFLA**	
Number of iterations	200		50		100	
Number of samples to be taken	NA		50		NA	
Optimum location of a bus	7		7		7	
Determined active power to be injected(kW)	613.9014		599.6		527	
Dg losses	31.7768		31.77		32.15	
Measured voltage/nominal voltage (p.u) of busses before & after active power injection	BEFORE	AFTER	BEFORE	AFTER	BEFORE	AFTER
BUS 1	1	1	1	1	1	1
BUS 2	0.989	0.994	0.989	0.994	0.989	0.993

BUS 3	0.973	0.985	0.973	0.985	0.973	0.983
BUS 4	0.961	0.979	0.961	0.979	0.961	0.977
BUS 5	0.95	0.976	0.95	0.975	0.95	0.972
BUS 6	0.941	0.974	0.941	0.973	0.941	0.962
BUS 7	0.934	0.974	0.934	0.973	0.934	0.969
BUS 8	0.932	0.973	0.932	0.972	0.932	0.967
BUS 9	0.931	0.972	0.931	0.971	0.931	0.967
BUS 10	0.931	0.972	0.931	0.971	0.931	0.966
BUS 11	0.956	0.974	0.956	0.974	0.956	0.971
BUS 12	0.954	0.973	0.954	0.972	0.954	0.97
BUS 13	0.954	0.972	0.954	0.971	0.954	0.969
BUS 14	0.953	0.972	0.953	0.971	0.953	0.969
BUS 15	0.953	0.971	0.953	0.971	0.953	0.969
BUS 16	0.949	0.975	0.949	0.974	0.949	0.971
BUS 17	0.94	0.972	0.94	0.97	0.94	0.968
BUS 18	0.939	0.971	0.939	0.971	0.939	0.967
BUS 19	0.937	0.97	0.937	0.969	0.937	0.965
BUS 20	0.937	0.969	0.937	0.969	0.937	0.965
BUS 21	0.936	0.969	0.936	0.968	0.936	0.964
BUS 22	0.931	0.972	0.931	0.971	0.931	0.966
BUS 23	0.93	0.971	0.93	0.97	0.93	0.965
BUS 24	0.929	0.97	0.929	0.969	0.929	0.964
BUS 25	0.929	0.97	0.929	0.969	0.929	0.964
BUS 26	0.929	0.97	0.929	0.969	0.929	0.964
BUS 27	0.931	0.972	0.931	0.971	0.931	0.966
BUS 28	0.931	0.972	0.931	0.971	0.931	0.966

Conclusion

From the above results tabulated in Cl 6.2.3(SFLA) it is understood that it takes a hundred iterations for finding the optimal DG size (527kW) required o the located week bus to ensure more than 0.95(P.U) voltage profile on all the busses connected to the network. It is perfectly locating the weakest node (6th or 7th) from 20 iterations onwards. It works very well in the selection of DG size even though it takes more iterations than the ABC method to converge.

From the point of more stable results and quick converging (50 iterations) Artificial BEE colony Algorithm overshadows the other two algorithms 200 iterations and 100 iterations respectively by Adaptive Particle Swarm Optimization (APSO) and Shuffled Frog Leap Algorithm (SFLA).

In the Sizing of DG area, the Shuffled Frog Leap Algorithm (SFLA) outperforms the other two methods. It recommended a highly optimal power injection size of 527kW, which is sufficient to maintain 95% of the nominal voltage (0.95 p.u.) across all buses in the distribution network. While Artificial BEE colony (ABC) and Adaptive Particle Swarm Optimization (APSO) respectively advise power injections of 600kW and 613kW. Therefore, using the SFLA method for the load flow analysis will be preferable from a commercial standpoint because it recommends the ideal DG size needed to enhance the voltage profile across all buses beyond 95% of the specified nominal value.

References

1. Alzubaidi, M., Hasan, K. N., Meegahapola, L., & Rahman, M. T. (2021). Identification of efficient sampling techniques for probabilistic voltage stability analysis of renewable-rich power systems. Energies, 14(8), 2328.
2. Song, Yue, David J. Hill, and Tao Liu. "Static voltage stability analysis of distribution systems based on network-load admittance ratio." IEEE Transactions on Power Systems 34.3 (2018): 2270-2280.
3. Ghaljehei, Mohammad, Ali Ahmadian, Masoud Aliakbar Golkar, Turaj Amraee, and Ali Elkamel. "Stochastic SCUC considering compressed air energy storage and wind power generation: A techno-economic approach with static voltage stability analysis." International Journal of Electrical Power & Energy Systems 100 (2018): 489-507.
4. Yao, R., Sun, K., Shi, D., & Zhang, X. (2018). Voltage stability analysis of power systems with induction motors based on holomorphic embedding. IEEE Transactions on Power Systems, 34(2), 1278-1288.
5. Zhang, M., Li, Y., Liu, F., Luo, L., Cao, Y., & Shahidehpour, M. (2017). Voltage stability analysis and sliding-mode control method for rectifier in DC systems with constant power loads. IEEE Journal of Emerging and Selected Topics in Power Electronics, 5(4), 1621-1630.
6. Aly, Mohamed M., and Mamdouh Abdel-Akher. "A robust quasi-direct continuous power-flow analysis using two-bus equivalents for voltage stability analysis of radial distribution systems." Electrical Engineering 100, no. 1 (2018): 47-57.
7. Xiao H, Li Y, Shi D, Chen J, Duan X. Evaluation of strength measure for static voltage stability analysis of hybrid multi-infeed DC systems. IEEE Transactions on Power Delivery. 2019 Feb 26;34(3):879-90.
8. Lee, Yunhwan, and Hwachang Song. "A reactive power compensation strategy for voltage stability challenges in the Korean power system with dynamic loads." Sustainability 11, no. 2 (2019): 326.
9. Ghaljehei, Mohammad, Ali Ahmadian, Masoud Aliakbar Golkar, Turaj Amraee, and Ali Elkamel. "Stochastic SCUC considering compressed air energy storage and wind power generation: A techno-economic approach with static voltage stability

analysis." International Journal of Electrical Power & Energy Systems 100 (2018): 489-507.

10. Yao R, Sun K, Shi D, Zhang X. Voltage stability analysis of power systems with induction motors based on holomorphic embedding. IEEE Transactions on Power Systems. 2018 Oct 16;34(2):1278-88.
11. Karthikeyan, K., and P. K. Dhal. "Multi verse optimization (MVO) technique based voltage stability analysis through continuation power flow in IEEE 57 bus." Energy Procedia 117 (2017): 583-591.