IJAMML 3:2 (2015) 163-175 December 2015
ISSN: 2394-2258
Available at http://scientificadvances.co.in
DOI: http://dx.doi.org/10.18642/ijamml_7100121602

NEW-FANGLED ALGORITHM FOR SOLVING REACTIVE POWER PROBLEM

K. Lenin[a], B. Ravindhranath Reddy[b] and M. Surya Kalavathi[c]

[a]Research Scholar, JNTU, Hyderabad 500085, India

[b]Deputy Executive Engineer, JNTU, Hyderabad 500085, India

[c]Professor, Department of Electrical and Electronics Engineering, JNTU, Hyderabad 500085, India

---

Abstract

In this paper, an Aeshnidae Algorithm (AA) is used to solve optimal reactive power problem. The key inducement of the aeshnidae algorithm (AA) instigate from static and dynamic swarming behaviours. These two swarming behaviours are very alike to the two key phases of optimization using meta-heuristics: exploration and exploitation. Aeshnidae generate sub swarms and fly over diverse areas in a static swarm, which is the key objective of the exploration phase. In the static swarm, however, aeshnidae fly in bigger swarms and along one direction, which is favourable in the exploitation phase. In this proposed aeshnidae algorithm, two essential phases of optimization, exploration and exploitation, are designed by modelling the social interaction of aeshnidae in navigating, searching for foods, and avoiding enemies when swarming dynamically or statistically. The projected aeshnidae algorithm (AA) has been tested in standard IEEE 30 bus test system and simulation results show clearly about the enhanced performance of the projected algorithm in tumbling the real power loss.

*Corresponding author.

E-mail address: gklenin@gmail.com (K. Lenin).

2010 Mathematics Subject Classification: 68W20.
Submitted by Li Li.
Received December 12, 2015

Keywords: aeshnidae algorithm, optimization, optimal reactive power, transmission loss.

1. Introduction

Different algorithms are utilized to solve the reactive power dispatch problem. Different types of numerical techniques like the gradient method [1, 2], Newton method [3], and linear programming [4-7] have been already used to solve the optimal reactive power dispatch problem. The voltage stability problem plays an important role in power system planning and operation [8]. Evolutionary algorithms such as genetic algorithm, hybrid differential evolution algorithm, biogeography based algorithm, a fuzzy based approach, an improved evolutionary programming [9-15] have been already utilized to solve the reactive power flow problem in [16-18] different methodologies are successfully handled the optimal power problem. In [19, 20], a programming based approach and probabilistic algorithm is used to solve the optimal reactive power dispatch problem. This paper proposes an aeshnidae algorithm (AA) to solve reactive power dispatch problem. Aeshnidae are measured as small predators that hound almost all other petite insects in environment. Fairy aeshnidae also predate on other marine insects and even small fishes. The attractive fact about aeshnidae is their exclusive and a typical swarming behaviour. Aeshnidae swarm for only two reasons: hunting and migration. The former is called static swarm, and the latter is called dynamic swarm. In static swarm, aeshnidae create small groups and fly back and forth over a small area to hunt other flying preys such as butterflies and mosquitoes [21-24]. Confined movements and rapid changes in the flying path are the key characteristics of a static swarm. In vibrant swarms, however, an enormous number of aeshnidae make the swarm for migrating in one direction over long distances. The key stimulation of the aeshnidae algorithm (AA) instigates from static and dynamic swarming behaviours. These two swarming behaviours are very alike to the two key phases of optimization using meta-heuristics: exploration and exploitation. Aeshnidae create sub-swarms and fly over different regions in a static swarm, which is the key objective of the exploration phase. In the static swarm, however, aeshnidae fly in bigger swarms and along one direction, which is constructive in the exploitation phase. The proposed AA has been evaluated in standard IEEE 30 bus test system. The simulation results show about the projected approach outperforms all the entitled reported algorithms in minimization of real power loss.

2. Objective Function

A. Active power loss

The objective of the reactive power dispatch problem is to minimize the active power loss and can be defined in equations as follows:

$$F = P_L = \sum_{k \in Nbr} g_k (V_i^2 + V_j^2 - 2V_i V_j \cos \theta_{ij}), \tag{1}$$

where F is objective function, P_L is power loss, g_k is conductance of branch, V_i and V_j are voltages at buses i, j, and Nbr is total number of transmission lines in power systems.

B. Voltage profile improvement

To minimize the voltage deviation in PQ buses, the objective function (F) can be written as

$$F = P_L + \omega_v \times VD, \tag{2}$$

where VD is voltage deviation, ω_v is a weighting factor of voltage deviation. The voltage deviation given by

$$VD = \sum_{i=1}^{Npq} |V_i - 1|, \tag{3}$$

where Npq is number of load buses.

C. Equality constraint

The equality constraint of the problem is indicated by the power balance equation as follows:

$$P_G = P_D + P_L, \tag{4}$$

where P_G is total power generation and P_D is total power demand.

D. Inequality constraints

The inequality constraint implies the limits on components in the power system in addition to the limits created to make sure system security. Upper and lower bounds on the active power of slack bus (P_g) and reactive power of generators (Q_g) are written as follows:

$$P_{gslack}^{\min} \le P_{gslack} \le P_{gslack}^{\max}, \tag{5}$$

$$Q_{gi}^{\min} \le Q_{gi} \le Q_{gi}^{\max}, \quad i \in N_g; \tag{6}$$

Upper and lower bounds on the bus voltage magnitudes (V_i) is given by

$$V_i^{\min} \le V_i \le V_i^{\max}, \quad i \in N; \tag{7}$$

Upper and lower bounds on the transformers tap ratios (T_i) is given by

$$T_i^{\min} \le T_i \le T_i^{\max}, \quad i \in N_T; \tag{8}$$

Upper and lower bounds on the compensators (Q_c) is given by

$$Q_c^{\min} \le Q_c \le Q_C^{\max}, \quad i \in N_C, \tag{9}$$

where N is the total number of buses, N_g is the total number of generators, N_T is the total number of transformers, and N_c is the total number of shunt reactive compensators.

3. Aeshnidae Algorithm

According to Reynolds [25], the behaviour of swarms follows three primitive principles: Separation- refers to the static collision evading of the individuals from other individuals in the neighbourhood. Alignment - indicates velocity matching of individuals to that of other individuals in neighbourhood. Cohesion - refers to the propensity of individuals towards the centre of the mass of the neighbourhood. The main objective of any swarm is survival, so all of the individuals should be attracted towards food sources and distracted outward enemies. The behaviours are mathematically modelled as follows: The separation is calculated as follows [25]:

$$D_i = -\sum_{j=1}^{N} Z - Z_j, \tag{10}$$

where Z is the position of the current individual, Z_j shows the position j-th neighbouring individual, and N is the number of neighbouring individuals.

Alignment is calculated as follows:

$$G_i = \frac{\sum_{j=1}^{N} G_j}{N}, \tag{11}$$

where G_j shows the velocity of j-th neighbouring individual.

The cohesion is calculated as follows:

$$H_i = \frac{\sum_{j=1}^{N} Z_j}{N} - Z, \tag{12}$$

where Z is the position of the current individual N is the number of neighbourhoods, and Z_j shows the position j-th neighbouring individual.

Attraction towards a food source is calculated as follows:

$$E_i = Z^+ - Z, \tag{13}$$

where Z is the position of the current individual and Z^+ shows the position of the food source.

Distraction outwards an enemy is calculated as follows:

$$J_i = Z^- + Z, \tag{14}$$

where Z is the position of the current individual and Z^- shows the position of the enemy.

To update the position of artificial aeshnidae in a search space and simulate their movements, two vectors are considered: step (ΔZ) and position (Z). The step vector is analogous to the velocity vector in particle swarm optimization (PSO), and the aeshnidae algorithm is developed based on the framework of the PSO algorithm. The step vector can be defined as follows:

$$\Delta Z_{t+1} = (dD_i + gG_i + hH_i + eE_i + jJ_i) + w\Delta Z_t, \tag{15}$$

where d shows the separation weight, D_i indicates the separation of the i-th individual, g is the alignment weight, G_i is the alignment of i-th individual, h indicates the cohesion weight, H_i is the cohesion of the i-th individual, e is the food factor, eE_i is the food source of the i-th individual, j is the enemy factor, J_i is the position of enemy of the i-th individual, w is the inertia weight, and t is the iteration counter.

After calculating the step vector, the position vectors are calculated as follows:

$$Z_{t+1} = Z_t + \Delta Z_{t+1}, \tag{16}$$

where t is the current iteration.

In a static swarm, however, alignments are very low while cohesion is high to assail preys. Consequently, we assign aeshnidae with high alignment and low cohesion weights when exploring the search space and low alignment and elevated cohesion when exploiting the search space. For conversion between exploration and exploitation, the radii of neighbourhoods are augmented proportional to the number of iterations. Another way to balance exploration and exploitation is to adaptively tune the swarming factors (d, g, h, e, j, and w) during optimization.

To perk up the arbitrariness, stochastic behaviour, and exploration of the artificial aeshnidae, they are requisite to fly around the explore space using an arbitrary walk (Levy flight) when there is no neighbouring solutions. In this case, the location of aeshnidae is modernized by using the following equation:

$$Z_{t+1} = Z_t + Levy(k) \times Z_t, \tag{17}$$

where t is the current iteration and k is the dimension of the position vectors.

The Levy flight is calculated as follows:

Levy flight [26] is a rank of non-Gaussian random processes whose arbitrary walks are drawn from Levy stable distribution. This allocation is a simple power-law formula $L(s) \sim |s|^{-1-\beta}$, where $0 < \beta < 2$ is an index. Mathematically exclamation, an easy version of Levy distribution can be defined as

$$L(s, \gamma, \mu) = \begin{cases} \sqrt{\frac{\gamma}{2\pi}} & \exp\left[-\frac{\gamma}{2(s-\mu)}\right]\frac{1}{(s-\mu)^{3/2}} \text{ if } 0 < \mu < s < \infty, \\ 0 \quad \text{if } s \leq 0, & \end{cases} \tag{18}$$

where $\gamma > 0$ parameter is scale (controls the scale of distribution) parameter, μ parameter is location or shift parameter. In general, Levy distribution should be defined in terms of Fourier transform

$$F(k) = \exp\left[-\alpha|k|^{\beta}\right], \quad 0 < \beta \leq 2, \tag{19}$$

where α is a parameter within $[-1, 1]$ interval and known as scale factor. An index of o stability $\beta \in [0, 2]$ is also referred to as Levy index. In particular, for $\beta = 1$, the integral can be carried out analytically and is known as the Cauchy probability distribution. One more special case when $\beta = 2$, the distribution correspond to Gaussian distribution. β and α parameters take a key part in determination of the distribution. The parameter β controls the silhouette of the probability distribution in such a way that one can acquire different shapes of probability distribution, especially in the tail region depending on the parameter β. Thus, the smaller β parameter causes the distribution to make longer jumps since there will be longer tail. It makes longer jumps for smaller values whereas it makes shorter jumps for bigger values.

The aeshnidae algorithm (AA) algorithm starts optimization process by generating a set of arbitrary solutions for a given optimization problems. In fact, the position and step vectors of aeshnidae are initialized by random values defined within the lower and upper bounds of the variables. In each iteration, the position and step of each aeshnidae are updated by using Equations (15)-(17). For updating Z and ΔZ vectors, neighbourhood of each aeshnidae is chosen by calculating the Euclidean distance between all the aeshnidae and selecting N of them. The position updating process is sustained iteratively until the end criterion is satisfied.

Initialize the Aeshnidae population Z_i $(i = 1, 2, \dots, n)$

Initialize step vectors ΔZ_i $(i = 1, 2, \dots, n)$

While the end condition is not satisfied

Compute the objective values of all aeshnidae

Update the food source and enemy

Modernize factors (d, g, h, e, j, and w)

Compute D, G, H, E, and J using Equations (10)-(11)

Modernize neighbouring radius if anaeshnidae has at least one neighbouring aeshnidae

Modernize velocity vector using Equation (15)

Modernize position vector using Equation (16)

Else

Modernize position vector using Equation (17)

End if

Ensure and correct the new-fangled positions based on the boundaries of variables

End while.

4. Simulation Results

Aeshnidae algorithm (AA) has been tested in IEEE 30-bus, 41 branch system. The system has 6 generator-bus voltage magnitudes, 4 transformer-tap settings, and 2 bus shunt reactive compensators. Bus 1 is considered as slack bus and 2, 5, 8, 11, and 13 are considered as *PV* generator buses and the other buses are taken as *PQ* load buses. Generators buses (*PV*) are 2, 5, 8, 11, 13 and slack bus is 1. Control variables limits are listed in Table 1. The power limits generators buses are displayed in Table 2. Table 3 shows the projected approach succeeded in keeping the control variables within limits. Table 4 narrates about the performance of the proposed AA algorithm. Table 5 summarize the comparison results of the optimal solution obtained by various standard methods.

Table 1. Basic variable limits (PU)

List of variables	Min. value	Max. value	Category
Generator Bus	0.90	1.08	Continuous
Load Bus	0.90	1.01	Continuous
Transformer-Tap	0.91	1.00	Discrete
Shunt Reactive Compensator	− 0.10	0.30	Discrete

Table 2. List of generators power limits

Bus	Pg.	Pg min	Pg max	Qg min
1	90.00	47	121	− 20
2	82.00	18	75	− 20
5	50.00	10	41	− 11
8	20.00	10	32	− 13
11	20.00	10	19	− 10
13	20.00	11	35	− 13

Table 3. Control variables values after optimization

Control variables	AA
V1	1.0605
V2	1.0512
V5	1.0319
V8	1.0402
V11	1.0811
V13	1.0609
T4, 12	0.00
T6, 9	0.01
T6, 10	0.90
T28, 27	0.90
Q10	0.11
Q24	0.11
Real power loss	4.2779
Voltage deviation	0.9050

Table 4. Performance of AA algorithm

Iterations	28
Time taken (secs.)	4.32
Real power loss	4.2779

Table 5. Comparison of real power loss

Methods	Real power loss (MW)
SGA (26)	4.98
PSO (27)	4.9262
LP (28)	5.988
EP (28)	4.963
CGA (28)	4.980
AGA (28)	4.926
CLPSO (28)	4.7208
HSA (29)	4.7624
BB-BC (30)	4.690
AA	4.2779

5. Conclusion

In this paper, aeshnidae algorithm (AA) has been efficiently solved the optimal reactive power dispatch problem. The projected algorithm has been tested in standard IEEE 30 bus system. Simulation study shows the robustness of projected aeshnidae algorithm (AA) method in providing improved optimal solution by decreasing the real power loss. The control variables values obtained after the optimization aeshnidae algorithm (AA) is well within the limits.

References

[1] O. Alsac and B. Scott, Optimal load flow with steady state security, IEEE Transaction. PAS (1973), 745-751.

[2] K. Y. Lee and Y. M. Paru and J. L. Oritz, A united approach to optimal real and reactive power dispatch, IEEE Transactions on Power Apparatus and Systems PAS 104 (1985), 1147-1153.

[3] A. Monticelli, M. V. F. Pereira and S. Granville, Security constrained optimal power flow with post contingency corrective rescheduling, IEEE Transactions on Power Systems: PWRS 2(1) (1987), 175-182.

[4] N. Deeb and S. M. Shahidehpur, Linear reactive power optimization in a large power network using the decomposition approach, IEEE Transactions on Power System 5(2) (1990), 428-435.

[5] E. Hobson, Network cons rained reactive power control using linear programming, IEEE Transactions on power systems PAS 99(4) (1980), 868-877.

[6] K. Y. Lee, Y. M. Park and J. L. Oritz, Fuel −cost optimization for both real and reactive power dispatches, IEE Proc. 131C(3) (1992), 85-93.

[7] M. K. Mangoli and K. Y. Lee, Optimal real and reactive power control using linear programming, Electr. Power Syst. Res. 26 (1993), 1-10.

[8] C. A. Canizares, A. C. Z. de Souza and V. H. Quintana, Comparison of performance indices for detection of proximity to voltage collapse, 11(3) (1996), 1441-1450.

[9] S. R. Paranjothi and K. Anburaja, Optimal power flow using refined genetic algorithm, Electr. Power Compon. Syst. 30 (2002), 1055-1063.

[10] D. Devaraj and B. Yeganarayana, Genetic algorithm based optimal power flow for security enhancement, IEE Proc.-Generation, Transmission and Distribution 152(6) (2005).

[11] A. Berizzi, C. Bovo, M. Merlo and M. Delfanti, A GA approach to compare ORPF objective functions including secondary voltage regulation, Electric Power Systems Research 84(1) (2012), 187-194.

[12] C.-F. Yang, G. G. Lai, C.-H. Lee, C.-T. Su and G. W. Chang, Optimal setting of reactive compensation devices with an improved voltage stability index for voltage stability enhancement, International Journal of Electrical Power and Energy Systems 37(1) (2012), 50-57.

[13] P. Roy, S. Ghoshal and S. Thakur, Optimal VAR control for improvements in voltage profiles and for real power loss minimization using biogeography based optimization, International Journal of Electrical Power and Energy Systems 43(1) (2012), 830-838.

[14] B. Venkatesh, G. Sadasivam and M. Khan, A new optimal reactive power scheduling method for loss minimization and voltage stability margin maximization using successive multi-objective fuzzy LP technique, IEEE Transactions on Power Systems 15(2) (2000), 844-851.

[15] W. Yan, S. Lu and D. Yu, A novel optimal reactive power dispatch method based on an improved hybrid evolutionary programming technique, IEEE Transactions on Power Systems 19(2) (2004), 913-918.

[16] W. Yan, F. Liu, C. Chung and K. Wong, A hybrid genetic algorithm interior point method for optimal reactive power flow, IEEE Transactions on Power Systems 21(3) (2006), 1163-1169.

[17] J. Yu, W. Yan, W. Li, C. Chung and K. Wong, An unfixed piecewise optimal reactive power-flow model and its algorithm for ac-dc systems, IEEE Transactions on Power Systems 23(1) (2008), 170-176.

[18] F. Capitanescu, Assessing reactive power reserves with respect to operating constraints and voltage stability, IEEE Transactions on Power Systems 26(4) (2011), 2224-2234.

[19] Z. Hu, X. Wang and G. Taylor, Stochastic optimal reactive power dispatch: Formulation and solution method, International Journal of Electrical Power and Energy Systems 32(6) (2010), 615-621.

[20] A. Kargarian, M. Raoofat and M. Mohammadi, Probabilistic reactive power procurement in hybrid electricity markets with uncertain loads, Electric Power Systems Research 82(1) (2012), 68-80.

[21] J. H. Thorp and D. C. Rogers, Thorp and Covich's Freshwater Invertbrates: Ecology and General Biology, Elsevier, Amsterdam, 2014.

[22] M. Wikelski, D. Moskowitz, J. S. Adelman, J. Cochran, D. S. Wilcove and M. L. May, Simple rules guide dragonfly migration, Biol. Lett. 2 (2006), 325-329.

[23] W. Russell, M. L. May, K. L. Soltesz and J. W. Fitzpatrick, Massive swarm migrations of dragonflies (Odonata) in eastern North America, Am. Midl. Nat. 140 (1998), 325-342.

[24] S. Mirjalili, Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Computing and Applications, in press (2015).

[25] C. W. Reynolds, Flocks, herds and schools: A distributed behavioral model, ACM SIGGRAPH Comput. Gr. 21 (1987), 25-34.

[26] Hüseyin Hakli and Harun Uğuz, Levy flight distribution for Scout bee in artificial bee colony algorithm, Lecture Notes on Software Engineering 1(3) (2013).

[27] Q. H. Wu, Y. J. Cao and J. Y. Wen, Optimal reactive power dispatch using an adaptive genetic algorithm, Int. J. Elect. Power Energy Syst. 20 (1998), 563-569.

[28] B. Zhao, C. X. Guo and Y. J. Cao, Multiagent-based particle swarm optimization approach for optimal reactive power dispatch, IEEE Trans. Power Syst. 20(2) (2005), 1070-1078.

[29] K. Mahadevan and P. S. Kannan, Comprehensive learning particle swarm optimization for reactive power dispatch, Applied Soft Computing 10(2) (2010), 641-652.

[30] A. H. Khazali and M. Kalantar, Optimal reactive power dispatch based on harmony search algorithm, Electrical Power and Energy Systems 33(3) (2011), 684-692.

[31] S. Sakthivel, M. Gayathri and V. Manimozhi, A nature inspired optimization algorithm for reactive power control in a power system, International Journal of Recent Technology and Engineering (IJRTE) 2(1) (2013), 29-33.

■