International Journal of Electrical and Computer Engineering (IJECE)
Vol. 9, No. 5, October 2019, pp. 3967~3974
ISSN: 2088-8708, DOI: 10.11591/ijece.v9i5.pp3967-3974

Swarm algorithms in dynamic optimization problem of reactive power compensation units control

V.Z. Manusov, P.V. Matrenin, N. Khasanzoda
Department of Industrial Power Supply Systems, Novosibirsk State Technical University, Russia

Article Info

Article history:

Received Oct 8, 2018
Revised Apr 16, 2019
Accepted Apr 29, 2019

Keywords:

Bees algorithm
Dynamic optimization
Operation control
Particle swarm optimization
Power-supply systems

ABSTRACT

Optimization of a power supply system is one of the main directions in power engineering research. The reactive power compensation reduces active power losses in transmission lines. In general, researches devoted to allocation and control of the compensation units consider this issue as a static optimization problem. However, it is dynamic and stochastic optimization problem that requires a real-time solution. To solve the dynamic optimization NP-hard problem, it is advisable to use Swarm Intelligence. This research deals with the problem of the compensation units power control as a dynamic optimization problem, considering the possible stochastic failures of the compensation units. The Particle Swarm Optimization and the Bees Algorithm were applied to solve it to compare the effectiveness of these algorithms in the dynamic optimization of a power supply system.

Corresponding Author:

Pavel V. Matrenin,
Department of Industrial Power Supply Systems,
Novosibirsk State Technical University,
20 Prospekt K. Marksa, Novosibirsk, 630073, Russia.
Email: pavel.matrenin@gmail.com

1. INTRODUCTION

Optimal distribution of reactive power of energy supply systems and in electrical networks is a quite topical and innovative task. The main reason for high attention towards the problem is the possibility without providing effective additional capital investments into equipment and other power system performance to achieve economies through solving the problem set. The efficiency of the compensation depends on the distribution of reactive power compensation units at power supply system nodes [1]. The optimization task of finding the optimal distribution and powers of reactive power compensation units (CU) is nonlinearity, no differentiability; it has multiple extremes and high computational complexity. Presently, the issues of finding solutions for similar problems are considered, based on artificial intelligence methods, such as Fuzzy logic, heuristic optimization methods, etc. Research [2] proposes the regulation of STATCOM using Fuzzy Logic concept. It allows tuning controllers to achieve a high efficiency of power grid operation. Fuzzy logic is a powerful mathematical tool for power system control [2, 3] but applying Fuzzy Logic requires expert knowledge of the system and its functioning. Therefore, developing fuzzy rules can be very time-consuming. It is much easier to use heuristic optimization methods that do not require special labor-intensive modifications. The Tabu search used in [4] and [5]. Research [5] also considers the Simulated Annealing algorithm. Now the Genetic algorithm is the most widely used optimization method in power supply system optimization [5-8]. The other population-based Swarm Intelligence algorithms are used successfully for the optimization of reactive power control, reactive power dispatch, and other power system optimization problems: Particle Swarm optimization [9-11], Fire Fly optimization [12]; Crow search [13], Cuckoo search [14], Bee algorithm [15].

This paper does not consider the placement of CUs in the network nodes; it considers the problem of the operational control of CU powers to ensure minimum active power losses while maintaining the stability

of the system. In networks, loads may change dynamically and equipment failures may occur, so it is necessary that CUs should be controlled in real time. For optimization problems with dynamically changing conditions, it is necessary to apply algorithms capable, first, of self-organization that is automatically adapting to the task and, secondly, can give efficient solutions fast enough to work in real time. Stochastic methods are especially productive for optimization problems that have such features as nonlinearity, non-differentiability, and high computational complexity, for stochastic and dynamic tasks [16-18]. Often it is impossible to make an analytical expression for them because the model has a complex structure and interconnections between its elements. It also leads to great laboriousness of calculation and complex topology of the search space.

This research considers the PSO and the BA for dynamic optimization in the dynamic optimization of CUs. At this stage of the research, the easiest option for the experiments associated with failures of CUs has been selected. It is the main aspect of the reliability of power systems [19]. The purpose of this stage is to analyze the dynamic properties of these algorithms. We strike to create modifications of these algorithms, based on this analysis for their further use under the conditions when, apart from the equipment failures, the powers of consumers of electricity in the grid are varied according to the certain distribution function.

2. RESEARCH METHOD

2.1. The power supply system considered

The power supply system 400 V of a production plant is considered. The power supply system represents a substation comprising four sections, and the arrangement of each section is radial. For the power distribution among consumers, power distribution points fed from the substation section bus bars are applied. Ten most congested distribution points are considered in each section (distribution cabinets), which provide power for asynchronous motors of the ventilation system, electric motors of pumps (the power of each is about 150 kW) and the main supply lines of boost compressors. Each main supply line provides from 10 to 15 compressors with 9 kW induction motors which are supplied by a stub.

2.2. The Mathematical model

The mathematical model of the considered optimization problem has the vector of CU powers (Q_{CU}) as control variables and criteria: active power losses → min & stable operation of the system → true. The second criterion we use as the penalty function $G(Q_{CU})$. As a result, the mathematical formulation can be written as follows:

$$\Delta P(Q_{CU}) + G(Q_{CU}) \rightarrow \min, \quad (1)$$

$$Q_{CU}\{Q_1, Q_2, \ldots, Q_n\} \quad (2)$$

$$0 < Q_i < Q_{max\,i}\,,\ i = 1, \ldots, n \quad (3)$$

ΔP (Q_{CU}) is the total losses of active power in power supply system.
Q_{CU} is the CU power vector.
Q_i is the CU power in the i-th node (if $Q_{max\,I} = 0$, then a CU is not installed in the i-th node);
n is the number of node where CUs are installed ($n = 14$ for considered power supply system).
$Q_{max\,i}$ is the maximum allowable CU power value in the i-th node.

2.3. Swarm intelligence algorithms

Term "Population-based" means that these algorithms use a "population" as a system of particles (agents). As a rule, such algorithms are based on natural mechanisms, such as evolutionary selection, cooperate activity of insects, etc. The term "particle" means a point in the search space mathematically, so it is some solution of an optimization problem. The main feature of the population-based algorithms is their self-organization. It ensures the algorithms' ability to explore the search space regardless of the space's dimension and topology. It provides the flexibility and possibility to find quasi-optimal solutions quickly. Population-based optimization algorithms can be split into Evolutionary algorithms and Swarm Intelligence (SI) algorithms. The evolutionary process is based on creating new populations at every new step concerning the experience obtained by the previous populations. Genetic algorithm, Genetic programming, Differential evolution are examples of Evolutionary algorithms. It uses principles of natural selection as inheritance, mutation, crossover. The word "swarm" can be defined as a group of objects or an aggregation of moving bodies. Swarming means movements of the particles in the search space, using some rules and indirect exchange of data between the particles through shared memory. In contrast to Evolutionary algorithms,

the particles are not created and destroyed, and the swarm population has no centralized control system. Swarm Intelligence algorithms should be considered as a particular group amid the population-based optimization algorithms since they share the same characteristic idea, based on the collective movements of decentralized particles and the indirect information exchange.

The application of the system approach allows isolating elements of an object and relationships between them. Also, it needs to define connections between an object and the outside world (inputs and outputs). An SI algorithm is specified certain data structures and operations on them like many other algorithms. SI uses the swarm, i.e., a population of particles as a base data structure and rules of moving particles as an algorithm of transformation this data. A distinctive feature of the SI algorithms is using some means of indirect information exchange between particles. The Particle Swarm Optimization uses the best-found position in the search space; the Ant Colony Optimization uses a pheromone graph; the Monkey Search algorithm uses a weighted center of particles' positions, etc. Finally, it is necessary to distinguish the means of data exchange of an SI algorithm and the outside world (optimization task). Thus, we can formulate basic parts of SI algorithm:

- Set of particles *S*.
- Mean of indirect interaction *M*.
- Algorithm of particles' moving, and interaction with optimization problem *A*.
- Parameters of heuristic rules *P*.
- Input to obtain data from the optimization problem being solved *I*.
- Output to send solutions of the optimization problem and the final algorithm result *O*.

2.4. Particle Swarm Optimization

The Particle Swarm Optimization algorithm was first proposed by J. Kennedy and R. Eberhart in 1995 [20]. Then it was improved by Kennedy, Eberhart, and Shi [21]. PSO is based on a bird flocks' behavior. Every bird (particle) coordinates own movements with the movements of whole flocks. In the PSO algorithm, every particle is denoted by a position vector, a velocity vector, and a value of the criterion. The vectors of position and velocity of all particles are updated according to the rules taking into account the best position of a particle, and the best position of the whole swarm. Also, the algorithm uses inertia weights of the particles, velocities restriction and the stochastic deviations. According to the scheme of the swarm algorithm description [15], the PSO algorithm may be represented by a tuple $\{S, M, A, P, I, O\}$.

a. A set of particles $S = \{S_1, S_2, \ldots, S_{|S|}\}$, $|S|$ is number of particles. At j-th iteration i-th particle is characterized by the state $S_{ij}=\{X_{ij}, V_{ij}, X^{best}_{ij}\}$, where $X_{ij}=\{X^1_{ij}, X^2_{ij}, \ldots, X^n_{ij}\}$ is the variable parameter vector (particle position), $V_{ij} =\{V^1_{ij}, V^2_{ij}, \ldots, V^n_{ij}\}$ is the velocity vector, $X^{best}_{ij}=\{b^1_{ij}, b^2_{ij}, \ldots, b^n_{ij}\}$ are the best (by value) fitness-functions of the particle position among all the positions it took during the algorithm operation from the 1st to the j-th iterations, l is the number of variable parameters.
b. A mean of indirect exchange is vector $M = X_j^{best}$ is the best value of the variable parameters vector derived among all particles from the 1st to the j-th iterations of the algorithm.
c. Algorithm *A* describes the steps of the PSO algorithm.

1) Generation of initial population (iteration number $j = 1$):

$$X_{i1} \leftarrow \text{random}(0,1),\ i=1, \ldots, |S|;$$
$$V_{i1} \leftarrow \text{random}(-\beta, \beta),\ i=1, \ldots |S|;$$
$$X_{i1}^{best} \leftarrow X_{ij},\ i = 1, \ldots |S|.$$

where random(0,1) is the vector of random numbers with dimensionality l (dimensionality of solution search space) uniformly distributed from 0 to 1.

2) Calculation of fitness-functions. The criterion calculation takes place in the mathematical model of the problem where X_{ij} vectors are entered from algorithms, and the results are returned to the algorithm through the interface $\{I, O\}$.

$$\text{if } f(X_{ij}) < f(X_{ij}^{best}) \text{ then } X_{ij}^{best} \leftarrow X_{ij},\ i = 1, \ldots, |S|;$$
$$\text{if } f(X_{ij}) < f(M) \text{ then } M \leftarrow X_{ij},\ i = 1, \ldots, |S|.$$

3) Particles' movement with respect to the tolerance region and to the velocity limitation:

$$V_{ij+1} \leftarrow V_{ij}\omega + \alpha_1(X_{ij}^{best} - Xij)\text{random}(0,1) + \alpha_2(M - X_{ij})\text{random}(0,1),\ i=1, \ldots, |S|, \quad (4)$$
$$\text{if } V_{ij+1} > \beta \text{ then } V_{ij+1} \leftarrow \beta,\ i = 1, \ldots, |S|,$$
$$\text{if } V_{ij+1} < -\beta \text{ then } V_{ij+1} \leftarrow -\beta,\ i = 1, \ldots, |S|.$$

where α_1, α_2, ω, V_{max} are algorithm parameters.
$X_{ij+1} \leftarrow X_{ij}+V_{ij+1}$, i=1, …, $|S|$.
if X_{ij+1} >1 then $X_{ij+1} \leftarrow 1$, i=1, …, $|S|$;
if $X_{ij+1} < 0$ then $X_{ij+1} \leftarrow 0$, i=1, …, $|S|$.

4) If at the j-th iteration a stop-condition is satisfied, then the value M is transmitted to output O or the transition to iteration 3.2 takes place.

d. Vector $P = \{\alpha_1, \alpha_2, \omega, \beta\}$ comprises the coefficients of algorithm A which influences the particles' movement in the search space. Coefficients α_1 and α_2 define the degree of accounting the individual and group experience of the particles, respectively. Coefficient ω characterizes inertial properties of the particles, and coefficient β defines limitations for the maximum velocity.

2.5. Bees Algorithm

The Bees Algorithm (not the Artificial Bee Colony Optimization [22]) was researched and developed by D.T. Pham et al. in 2005 [23]. It is based on the simulation of the behavior of bees in their searching for nectar and the indirect exchange of information between bees. Bee swarm sends several scouts in random directions to search for nectar. Returning, scouts report on the areas found in the field with flowers containing nectar, and on them fly out the other bees [15]. In this case, the more on the site of nectar, the more bees go to it. However, the bees can randomly deviate from the chosen direction. After the return of all the bees in the hive, information exchange and sending of bees again.

According to the description scheme of swarm algorithms [15], the BA may be represented by a tuple $\{S, M, A, P, I, O\}$.

a. A set of particles (bees) $S = \{S_1, S_2, \ldots, S_{|S|}\}$. At the j-th iteration the i-th particle is characterized by the state $S_{ij}=\{X_{ij}\}$, where $X_{ij} = X^1_{ij}, X^2_{ij}, \ldots, X^n_{ij}$ is the variable parameters vector (the particle position), l is the dimensionality of the solution search space.
b. Means of indirect exchange M is a list of the best and perspective positions found in the j-th iteration, $M = \{N^b_{ij}, N^g_{kj}\}$, i=1, ..., n^b, k = 1, ..., n^g.
c. Algorithm A describes the steps of the BA.

1) Generation of initial population (j = 1) is fulfilled only for a subset of scout particles:

$$X_{i1} \leftarrow \text{random}(0,1),\ i = 1, \ldots, n^s$$

where n^s is the number of scout particles. Other particles are considered as inactive this time (only at the first iteration).

2) Calculation of fitness-functions. The criterion calculation takes place in the mathematical model of the problem where X_{ij} vectors are entered from algorithms, and the results are returned to the algorithm through the interface $\{I, O\}$.

$$\text{if } f(X_{ij}) < f(X_{ij}^{best}) \text{ then } X_{ij}^{best} \leftarrow X_{ij},\ i = 1, \ldots, |S|.$$

3) Particles' movement. Among all particles, n^b particles with the best values of target function are chosen, and then, in the rest of the set, n^g particles with the best values are chosen. The lists of the best and perspective positions $M = \{N^b_{ij}, N^g_{kj}\}$ are generated using the chosen particles. Herewith, the distance between any two positions in M over each coordinate in the solution search space must be not less than the values of parameter rx. Worker particles are sent to the vicinity of these positions c^b particles are sent to the vicinity of each best position and c^g particles are sent to the vicinity of each perspective position. Thus, the positions of all worker particles are determined as follows:

$$X_{(i-1)cb + kj} \leftarrow N^b_{ij-1} + \text{random}(-1, 1) \cdot rad,\ i = 1,\ldots, n^b,\ k = 1,\ldots,c^b;$$
$$X_{nbcb + (i-1)cb + kj} \leftarrow N^b_{ij-1} + \text{random}(-1, 1)\cdot \text{rad},\ i = 1,\ldots, n^g,\ k = 1,\ldots,c^g.$$

where n^s, n^b, n^g, c^b, c^g, rad are the parameters of the algorithm.
In this case, scout particles are sent to random positions the coordinates of which are random values uniformly distributed in the tolerance range:

$$X_{nb\cdot cb+ng\cdot cg+ij} \leftarrow \text{random}(-1, 1) \cdot rad,\ i=1,\ldots, n^s.$$

Finally, the limitation of search space is checked:

if $X_{ij+1} > 1$ then $X_{ij+1} \leftarrow 1$, $i = 1, \ldots, |S|$
if $X_{ij+1} < 0$ then $X_{ij+1} \leftarrow 0$, $i = 1, \ldots, |S|$

4) If at the j-th iteration a stop-condition is satisfied, then the value is transmitted to output O, or the transition to iteration 3.2 takes place.

d. Algorithm parameters used in this description form vector $P = \{n^s, n^b, n^g, c^b, c^g, rad, rx\}$. The coefficient *rad* defines particle scattering in sending to the best and prospective positions. The coefficient *rx* defines the minimum possible distances between these positions. The value for the expression $n^s + n^b c^b + n^g c^g$ is equal to the total number of the swarm particles ($|S| = n^s + n^b c^b + n^g c^g$). The selection of the algorithm parameters heavily affects the quality of derived solutions, so to increase the algorithm efficiency, it is necessary to adapt parameters.

2.6. Application of the swarm algorithms to the problem considered

In the descriptions of the SI algorithms, we used constraints of the search space from 0 to 1 for each dimension. The introduced restrictions are necessary for successful adaptation and simple integration of the algorithm for solving various optimization problems. The fact is that some SI parameters have a different effect depending on the scale of optimization variables. Therefore, if a range of search space is from 0 to 100, the distance from the middle (50) to the high edge (100) will be 75. While if this distance is scaled from 0 to 1, then the same distance will be equal to 0.75 and the degree of attraction between the particles would be completely different. At the same time, in fact, this distance is equivalent to the optimization task. Another example is the radius of the scatting area (rad) in BA. Obviously, the larger the distance between the boundaries of a search space direction, the greater should be the radius of the scattering area. Moreover, for a non-homogeneous search space, it would be necessary to introduce different values of the parameters for different directions.

As a result, the algorithm parameters would have to be changed even in the case of a simple change of measurement units in the task (hours to seconds, kilometers to feet, etc.). To avoid this, we suggest specifying the search space bounded from 0.0 to 1.0 in all directions into the algorithm. And mapping of the particle's coordinate from the algorithm into the values of the optimized variables being solved. In the simplest variant, this can be done as follows. We denote the coordinate vector of a particle $X=\{X_1, X_2,\ldots, X_L\}$, and the vector of optimized variables $Q = \{Q_1, Q_2,\ldots, Q_L\}$. Under the restrictions $\alpha_i \le q_i \le b_i$, we obtain a mapping: $q_i=x_i(b_i - \alpha_i) + \alpha_i$, $i = 1, \ldots, L$.

For the interaction of the algorithms with the model of the optimization problem discussed, it is necessary to set up a correspondence between position X of a particle and the vector specifying the CU arrangement in the grid (Q_{CU}). Vector X is used not as the CU power vector, but as the coefficients' vector, so the power of each CU was determined as the product of the X vector element and the calculated maximum allowable power of the CU in the corresponding node: $Q_i = X_i Q_{maxi}$

2.7. The experiment

The experiment simulated the failure cases of one of the CU in the grid. It is necessary to carry out a real-time automatic adjustment of powers of another CU to preserve the stability of the system and minimize the active power losses. There are two alternative ways in the case of using the population-based algorithms for dynamically changing optimization problems.

a. "Restart". If conditions of an optimization task change, then an optimization algorithm runs again to solve this new task. So dynamically changing task is considered as a set of different static optimization tasks. On the one hand, it ignores the algorithm experience found by solving the task before condition changing. On the other hand, the algorithm can get out of a local extremum.
b. "Without restart". If conditions of an optimization task change, an optimization algorithm does not start from scratch, but from the current state. In this instance, there is a chance to quickly find an effective solution to the task modified, but there is a risk of getting stuck in a current local extremum.

During the experiments, each of the algorithms considered (PSO, BA) solved the optimization task (1-3) separately. The experiments for each algorithm were performed as follow:

a. Find the quasi-optimal solution X^* of task (1) when all the CUs are operational.
b. For each node i of the grid that has the CU ($i = 1,\ldots, n$).

1) Assume that the CU_i refused and ceased to function completely.
2) Continue the process of optimization of the algorithm for the task modified, i.e. use the restart. The process is stopped after 200 000 iterations after the conditions of the task have been changed.

3) Start the optimization algorithm from scratch without taking into account the previous solution, i.e. use the way without restart. The process is stopped after 200 000 iterations after the conditions of the task have been changed.

As the solution of the optimization problem is expected in real time, it is important how quickly a quasi-optimal solution to the task will be found after changing the conditions of the task. Therefore, the results were recorded at 100, 500, 1000 and 2000 iterations. The final results obtained after 200 000 iterations are rather of the theoretical value.

3. RESULTS AND DISCUSSION

The experimental total results averaged over all the CUs are shown in Table 1 and 2. The first column defines the algorithm used, the second one shows the way used for accounting changing conditions of the problem (with restart or without it). The following columns show the values of the optimization criterion (active power losses, kW) after the given number of algorithm iterations. Iterations were counted after changing the conditions of the problem, that is, from step 2.1. Table 1 shows the deviation of the values of the optimization criterion (ΔP) from the best value found among all the algorithms after 200000 iterations. The deviations are averaged over all the CUs. The maximum deviation is shown in Table 2. Figure 1 shows the information from Tables 1 and 2 correspondingly.

Table 1. Comparison of the algorithms' efficiency, the average deviation

Algorithm	Restart?	average deviation ΔP of the best, %				
		100	500	1000	2000	200000
PSO	restart	4.13	4.13	4.13	4.12	4.12
PSO	w/o restart	8.03	8.02	8.02	8.02	8.02
BA	restart	2.73	2.65	2.61	2.56	0.60
BA	w/o restart	2.62	2.55	2.52	2.48	0.49

Table 2. Comparison of the algorithms' efficiency, the maximum deviation

Algorithm	Restart?	maximum deviation ΔP of the best, %				
		100	500	1000	2000	200000
PSO	restart	12.57	12.57	12.57	12.57	12.57
PSO	w/o restart	25.17	25.173	25.17	25.17	25.17
BA	restart	4.07	3.98	3.94	3.88	2.07
BA	w/o restart	3.83	3.80	3.78	3.74	1.98

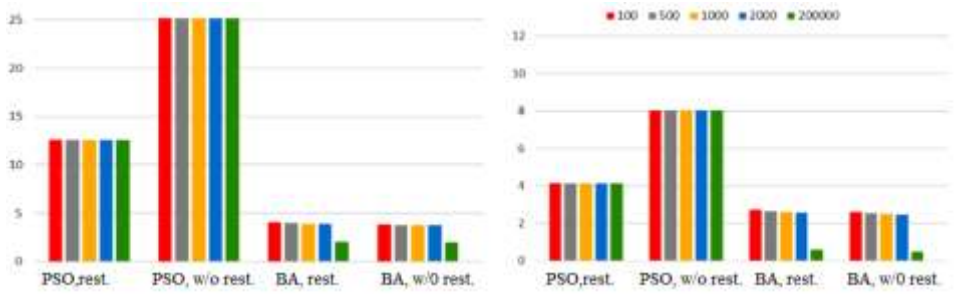

Figure 1. Comparison of the algorithms' efficiency, the average deviation (left), the maximum deviation (right)

The experiment shows that the BA provided the best solution, regardless of the number of iterations. The PSO with restart gave significantly better results than the PSO without restart. Moreover, the performance of the PSO without restart is hardly improved by increasing the number of iterations, which confirms that the PSO is unable to get out of a local extremum in case of changing the conditions of the task. It is attributed to the fact that when the PSO convergences in the neighborhood of an extremum, elements $(X_{ij}^{best} - X_{ij})$ and $(M - X_{ij})$ in Equation (4) appeared to be very small, and since the speeds of the particles are too small for them to leave the extremum in a reasonable time.

Currently, a number of limitations have been adopted in the work. The model of CU failures and their distribution is not used. Only the consequences of failures are considered. In the future, it is planned to complicate the optimization problem considered in the article and bring it closer to the real conditions. To do this, the model of the power supply system will include not only failures of CU, which, in fact, occur rarely, but also take into account the dynamic changes in loads in networks and operating modes of consumers.

4. CONCLUSION

The problem of control of reactive power sources in a power supply system is considered as a dynamic optimization problem solved in real-time. The criterion of the task is to reduce the loss of active power in the power supply system. As the controlled variables, the capacities of the reactive power compensation units are chosen. The compensation units' failures were simulated, leading to dynamic changes in the conditions of the problem. To solve the optimization problem, SI algorithms were used: the Particle Swarm Optimization algorithm and the Bees Algorithm. When solving dynamic optimization tasks, these algorithms can work either permanently, reacting to all changes in the conditions of the problem, or restart each time the conditions change. These two options can be designated "without restarting" and "with restart".

The experiments showed that for the Particle Swarm Optimization algorithm it is necessary to do a restart, otherwise, the algorithm does not come out of the local extremum since the particles of the Particle Swarm Optimization algorithm tend to collect in certain amount iterations in the vicinity of one extremum, refer Tables 1, 2. The perfect results were shown by the Bees Algorithm, while the option without restart was slightly more effective (by 0.1%). With correctly configured heuristic coefficients of the algorithm, the particles of the Bees Algorithm are always dispersed over several extremes, which allows both to take into account before found solutions and to find new ones. As the future work, it is planned to investigate the SI algorithms for problem of the optimal control in microgrids. For microgrids the operational control and reactive power sharing are really actual and important [24, 25].

ACKNOWLEDGEMENTS

The research was carried out under the State Assignment of the Ministry of Education and Science of the Russian Federation, Project 8.6809.2017/8.9.

REFERENCES

[1] J. W. Dixon and L. A. Moran, "Reactive power compensation technologies," *AccessScience*, 2018.
[2] A. Augustine, et al., "Voltage regulation of STATCOM using fuzzy self tuning PI controller," in *Circuit, Power and Computing Technologies (ICCPCT), Nagercoil*, pp. 1-7, 2016.
[3] F. C. Lu and Y. Y. Hsu, "Fuzzy dynamic programming approach to reactive power/voltage control in a distribution substation," *IEEE Transactions on Power Systems*, vol. 12, pp. 681-688, 1997.
[4] C. A. Rajan and M. R. Mohan, "An Evolutionary Programming Based Tabu Search Method for Solving the Unit Commitment Problem," *IEEE Transactions on Power Systems*, vol. 19, pp. 577-585, 2014.
[5] A. H. Mantawy, et al., "Integrating Genetic Algorithms, Tabu Search, and Simulated Annealing for the Unit Commitment Problem," *IEEE Transactions on Power Systems*, vol. 14, pp. 829-836, 1999.
[6] D. Dervani and J. P. Roselyn, "Genetic algorithm based reactive power dispatch for voltage stability improvement," *International Journal of Electrical Power & Energy Systems*, vol. 32, pp. 1151-1156, 2010.
[7] V. Z. Manusov, et al., "Implementation of Population Algorithms to Minimize Power Losses and Cable Cross-Section in Power Supply System," *International Journal of Electrical and Computer Engineering*, vol. 6, pp. 2955-2961, 2016.
[8] Md. I. Azim and Md. F. Rahman, "Genetic Algorithm Based Reactive Power Management by SVC," *International Journal of Electrical and Computer Engineering*, vol. 4, pp. 200-206, 2014.
[9] H. Yoshida, et al., "Particle swarm optimization for reactive power and voltage control considering voltage security assessment," *IEEE Transactions on Power System*, vol. 15, pp. 1232-1239, 2000.
[10] J. J. Jamian, et al., "A New Particle Swarm Optimization Technique in Optimizing Size of Distributed Generation," *International Journal of Electrical and Computer Engineering*, vol. 1, pp. 137-146, 2012.
[11] M. N. Dazahra, "Optimal Location of SVC using Particle Swarm Optimization and Voltage Stability Indexes," *International Journal of Electrical and Computer Engineering*, vol. 6, pp. 2581-2588, 2016.
[12] V. Z. Manusov, et al., "Firefly algorithm to optimal distribution of reactive power compensation units," *International Journal of Electrical and Computer Engineering*, vol. 8, pp. 1758-1765, 2018.
[13] M. Lakshmi and A. R. Kumar, "Optimal Reactive Power Dispatch using Crow search Algorithm," *International Journal of Electrical and Computer Engineering*, vol. 8, pp. 1423-1431, 2018.
[14] S. S. Reddy, "Optimal Reactive Power Scheduling Using Cuckoo Search Algorithm," *International Journal of Electrical and Computer Engineering*, vol. 7, pp. 2349-2356, 2017.

[15] V. Z. Manusov, et al., "Swarm intelligence algorithms for the problem of the optimal placement and operation control of reactive power sources into power grids," *International Journal of Design & Nature and Ecodynamics*, vol. 12, pp. 101-112, 2017.
[16] A. P. Engelbrecht, "Fundamentals of Computational Swarm Intelligence," John Wiley & Sons, pp. 672, 2005.
[17] L. Liu, et al., "Particle Swarm Optimization with Composite Particles in Dynamic Environments," *IEEE Transactions on Systems, Man, and Cybernetic*s, *Part B*, vol. 40, pp. 1634-1648, 2010.
[18] T. Blackwell and J. Branke, "Multiswarms, exclusion, and anti-convergence in dynamic environments," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 459-472, 2006.
[19] K. Wang, et al., "A Practical Structure and Control for Reactive Power Sharing in Microgrid," *IEEE Transactions on Smart Grid*, vol. 10, pp. 1880-1888, 2017.
[20] J. Kennedy and R. Eberhart, "Particle swarm optimization," *in IEEE International Conference on Neural Networks*, vol. 4, pp. 1942-1948, 1995.
[21] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," *in Congress on Evolutionary Computation, 27-30 May 2001; Seoul, South Korea. IEEE,* pp. 81-86, 2001.
[22] D. Karaboga, "An idea based on honey bee swarm for nuerical optimization," Erciyes University, Kayseri, Turkey, pp. 10, 2005.
[23] D. T. Pham, et al., "The Bees Algorithm – A Novel Tool for Complex Optimisation Problems," *Manufacturing Engineering Centre, Cardiff University, Cardiff*, 2005.
[24] J. He, et al., "An enhanced islanding microgrid reactive power imbalance power and harmonic power sharing scheme," *IEEE Transactions Power Electronics*, vol. 30, pp. 3389-3401, 2015.
[25] W. Qin, et al., "Reactive Power Aspects in Reliability Assessment of Power Systems," *IEEE Transactions on Power Systems*, vol. 26, pp. 85-92, 2011.

BIOGRAPHIES OF AUTHORS

Vadim Zinovievich Manusov received the B.S. and the PhD degrees electrical engineering from Novosibirsk Electric Technical Institute, Novosibirsk, Russia in 1963 and 1986, respectively. He is a Professor of the Department of Industrial Power Supply Systems in Novosibirsk State Technical University, Russia. His current research area is artificial intelligence technologies and probabilistic methods in electric power systems.

Pavel Viktorovich Matrenin received the B.S. and M.S. degrees information technologies from Novosibirsk State Technical University, Novosibirsk, Russia in 2012 and 2014, respectively. He is an assistant in Novosibirsk State Technical University. His main interests are related to stochastic optimization methods, design and development information systems, and artificial intelligence technologies in electric power systems.

Nasrullo Khasanzoda received the specialty degree electrical engineering from Tajik Technical University named after academician M.S. Osimi, Tajikistan in 2013. He is a PhD student in Novosibirsk State Technical University. His main interests are renewable energy and applying Artificial Intelligence to optimal control of renewable resources.