



NEW SWARM INTELLIGENCE BASED OPTIMIZATION ALGORITHMS FOR THE OPTIMIZATION OF MICROGRIDS

Tuba TANYILDIZI AĞIR^{*1}, *Zafer AYDOĞMUS*², *Bilal ALATAS*³

^{*1}Lecturer, University of Batman, Vocational High School Electronic Communication Technology Program

²Prof. Dr., University of Firat, Technology Faculty, Department of Electrical-Electronics Engineering

³Assoc. Prof. Dr., University of Firat, Engineering Faculty, Department of Software Engineering

* Corresponding author; E-mail: tanyildizituba@gmail.com

Received: 26 September 2018; Accepted: 22 November 2018

The need for new energy sources has increased due to reasons such as the development of technology, the increase in electricity demand, the decrease of fossil resources, and environmental pollution. Renewable energy sources are self-renewing, friendly, and clean energy sources. Microgrids are small power energy networks consisting of renewable and non-renewable energy sources, batteries, inverters, and loads. They can be operated connected to the network and independently from the network. Metaheuristic methods are algorithms that can achieve optimum results in the search space. In this study, optimization of a microgrid composed of a wind turbine, solar panel, diesel generator, inverter, and loads has been investigated with multi-objective hybrid metaheuristic algorithms. Optimization is aimed at reducing emissions, increasing reliability, and optimizing energy resources. Swallow Swarm Optimization (SSO) and Hybrid Particle Swallow Swarm Optimization (HPSSO) with different iterations and populations are compared for the first time.

Key words: *Microgrid, Metaheuristic algorithms, SSO, HPSSO.*

1. Introduction

Scientists have been in search of different power networks due to increasing social demands and insufficient existing power networks, and in order to improve reliability and quality [1]. For these reasons, significant changes have occurred in the electrical systems of developed countries. These changes have increased the use of renewable energy sources. However, in recent years, importance has been given to the use of renewable energy to ensure a broad liberalization of the electricity market and to preserve the quality of energy supplied to consumers [2].

Distributed production units have arisen due to environmental problems, market regulation, incentive policies, and the increase in global electricity demand. Distributed production units have many advantages such as increasing reliability, reducing power losses, improving power quality, and integration of renewable energy sources. However, these production units also have negative effects such as network security, system voltage, power system control, and power balance [3].

Renewable energy sources are often used as an alternative energy source in modern power systems. [4]. These energy sources are solar, wind, geothermal, biomass, and tidal. Many countries aimed at exceeding 20% of the power consumption used by renewable energy sources by 2020 [5]. As these energy sources have many advantages, these resources have some disadvantages such as proper working with other

production units. Some of the disadvantages caused by these sources are system voltage and frequency regulation, maintenance, and protection of renewable energy sources, network-dependent and grid-independent mode control arrangement [4].

Microgrid consists of distributed energy sources, storage units, inverter, and loads. These networks significantly increase the reliability of power systems in case of a power cut or if part of the network requires electricity. A well-designed distribution system can deal with load and systemic failures [6]. A microgrid can operate as connected to the network or independent of the network. Each distribution unit is connected to the microgrid via an interface inverter. The primary responsibility of the interface inverter is to regulate the frequency and amplitude of the output voltage, and thus to provide active and reactive power sharing. However, it can be used to improve the power quality of these inverters by selecting an appropriate control scheme [7]. A general power system uses battery energy storage units to avoid power outages and power fluctuations caused by environmental factors. Renewable energy is converted to DC and stored with energy storage elements and then converted to AC. This approach can easily adapt to existing electrical installations and accelerate the use of renewable energy [5].

Most engineering optimization problems are often difficult to solve, and many methods try to solve these complex problems. In these problems, the search space depends on the problem size. For this reason, traditional optimization methods do not provide a suitable solution for complex non-linear problems. For this reason, many metaheuristic algorithms are designed to solve such type of problems. Researchers have used metaheuristic algorithms to come up with complex problems such as timing problems, data clustering, image and video processing, and constructing neural networks [8]. The metaheuristic algorithms are global optimization methods that mimic natural phenomena and social behaviors. For example, biological evolution, thermal annealing, animal behavior, improvised music, swarming behavior, and so on. Two important features of these optimization methods are diversification and intensification. Diversification allows the optimizer to discover the search space more efficiently [9]. Intensification serves to explore the best available solutions and to select the best candidate designs. Specific objectives of the development of modern metaheuristic algorithms are to solve problems faster, to solve complex problems and to obtain more robust methods. Nevertheless, nature is a major source of inspiration to propose new metaheuristic approaches [10].

2. Swallow Swarm Optimization

There are three types of particles in this algorithm. The explorer particle, the aimless particle, and the leading particle. These particles move parallel to each other and are always in interaction. Every particle in the colony is responsible for directing the colony to a better state. Fig. 1 shows the particle types.

- **Explorer particle:**

The most important part of the colony is the explorer particles. Their primary responsibility is to make an exploration of space. The swallow emits a different tone of voice when the extract reaches the point of the candle and plays a role as chief leader [11].

- **Aimless particles:**

These particles are the worst particles when compared to other particles in the colony. The responsibilities of them in the group are randomly researching and exploration of the problem space. However, they have not any information about the position of the head leader and the local leader [12]. The equation of the update step for the aimless particle is shown in Eq. 1. In this equation, O_i^k represents the aimless particle, $rand$ is a random number between 0 and 1; min_s is the lower limit of variables, and max_s is the upper limit of the variables.

$$O_i^{k+1} = O_i^k + [rand(-1,1) * \frac{rand(min_s,max_s)}{1+rand()}] \quad (1)$$

- **Leading particles:**

These particles are called leaders in the SSO algorithm. Their location and quantities may vary at each level. These particles can be distributed or collected in the search space. The local leader is defined as the best leader of the sub-colonies. In the swallow, thousands of members colony are divided into a series of sub-colonies. These sub-colonies have a leader. This leader can be changed repeatedly by wiser and more powerful swallows. Lead swallow is the closest to food and resting places. The task of this leader is to guide other colony members [13]. The position update step of a principal leader is shown in equation Eq. 2. The acceleration coefficient update step equation is expressed in Eq. 3 and Eq. 4. Position update step equation and explorer particle update step equation are shown in Eq. 5 and Eq. 6 respectively. In these equations, VHL_i is velocity vector of head leader, α_{HL} is acceleration coefficient, e_{best} is the best value of the particle, e_i is the explorer particle, β_{HL} is used for acceleration coefficient, HL_i is a head leader, and VLL_i is local leader.

$$VHL_{i+1} = VHL_i + \alpha_{HL}rand()(e_{best} - e_i) + \beta_{HL}rand()(HL_i - e_i) \quad (2)$$

$$\left\{ \begin{array}{l} \alpha_{HL} = if(e_i = 0 || e_{best} = 0) \rightarrow 1.5 \\ \alpha_{HL} = if(e_i < e_{best}) \&\& ((e_i) < HL_i) \rightarrow \frac{rand()e_i}{e_i \times e_{best}} \quad e_i, e_{best} \neq 0 \\ \alpha_{HL} = if(e_i < e_{best}) \&\& ((e_i) > HL_i) \rightarrow \frac{2 \times rand()e_{best}}{1/(2 \times e_i)} \quad e_i \neq 0 \\ \alpha_{HL} = if(e_i > e_{best}) \rightarrow \frac{e_{best}}{1/(2 \times rand())} \end{array} \right. \quad (3)$$

$$\left\{ \begin{array}{l} \beta_{HL} = \{if(e_i = 0 || e_{best} = 0) \rightarrow 1.5 \\ \beta_{HL} = if(e_i < e_{best}) \&\& (e_i < HL_i) \rightarrow \frac{rand()e_i}{e_i \times HL_i} \quad e_i, HL_i \neq 0 \\ \beta_{HL} = if(e_i < e_{best}) \&\& (e_i > HL_i) \rightarrow \frac{2 \times rand()HL_i}{1/(2 \times e_i)} \quad e_i \neq 0 \\ \beta_{HL} = if(e_i > e_{best}) \rightarrow \frac{HL_i}{1/(2 \times rand())} \end{array} \right. \quad (4)$$

$$V_{i+1} = V_{HL_{i+1}} + V_{LL_{i+1}} \quad (5)$$

$$e_{i+1} = e_i + V_{i+1} \quad (6)$$

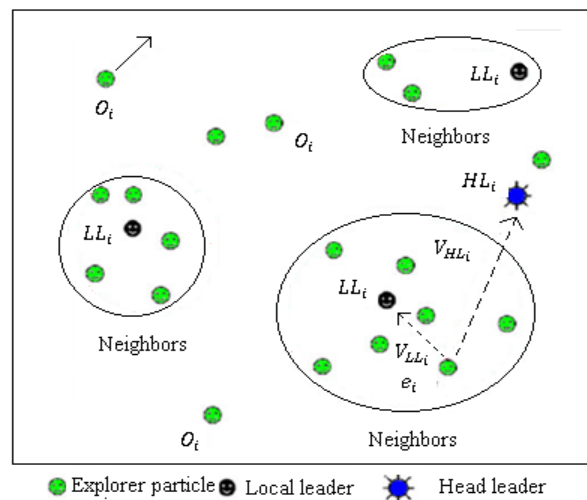


Figure 1. Types of particles [13]

3. Hybrid Particle Swallow Swarm Optimization (HPSSO)

HPSSO contains two key features of the SSO added to the basic PSO equation when specific sub-colonies and particles for specific tasks are considered. Similar to SSO, there are leaders (head leader, local leader), explorers and aimless particles. The size of the population is determined by the aimless particles and sub-colonies. The HPSSO starts with a random speed set and a random set of particles in the search area. The position and speed of each particle are constantly updated to search for the optimal solution. The best particle is identified as the head leader. The next particles are set from top to bottom in the position of local leaders. Aimless particles are selected from the worst individuals from top to bottom. The remaining particles are set as explorers. Each explorer particle occurs when the updated velocity vector is added to the current position of this particle. Compared to the PSO, it contains an additional term to describe the contribution of local leaders [14]. The position update step equation is shown as Eq. 7, and speed update step equation is shown as Eq. 8.

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (7)$$

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) + c_3 r_3 (P_{l(i)}^k - X_i^k) \quad (8)$$

$P_{l(i)}^k$ is the local leader of the lower colony of the i th particle, r_3 is a random number in the range (0,1). c_3 is the learning factor that controls proximity perception and ω is the inertia weight that controls the effect of the previous velocity.

In each iteration, the position of a particle within a sub-colony can be changed to move away from the current local leader and join another group. The distance between each explorer particle and a local leader is used to determine the sub-colony of each explorer particle in the group of the nearest local leader [15]. Distance update is shown in Eq. 9 and Eq. 10.

$$dist_{i,j} = |X_i - P_{i,j}|, j = 1, 2, \dots, N_{sub-colony} \quad (9)$$

$$dist_{i,j} = \sqrt{(X_i^1 - P_{i,j}^1)^2 + (X_i^2 - P_{i,j}^2)^2 + \dots + (X_i^{n_g} - P_{i,j}^{n_g})^2} \quad (10)$$

n_g is the number of design variables. $dist_{i,j}$ is the distance between the explorer particle i and the local leader j . Three possible options are taken for aimless particles.

- As in SSO, they only make a random search.
- They perform a local search in the area of the local leader.
- They dynamically search within the region where the global leader is located.

If option two is chosen, the number of aimless particles coincides with the number of sub-colonies, and aimless particles may not identify each sub-colony. In this case, the distance between the worst particle and the local leaders is the criterion for the assignment of the aimless particle.

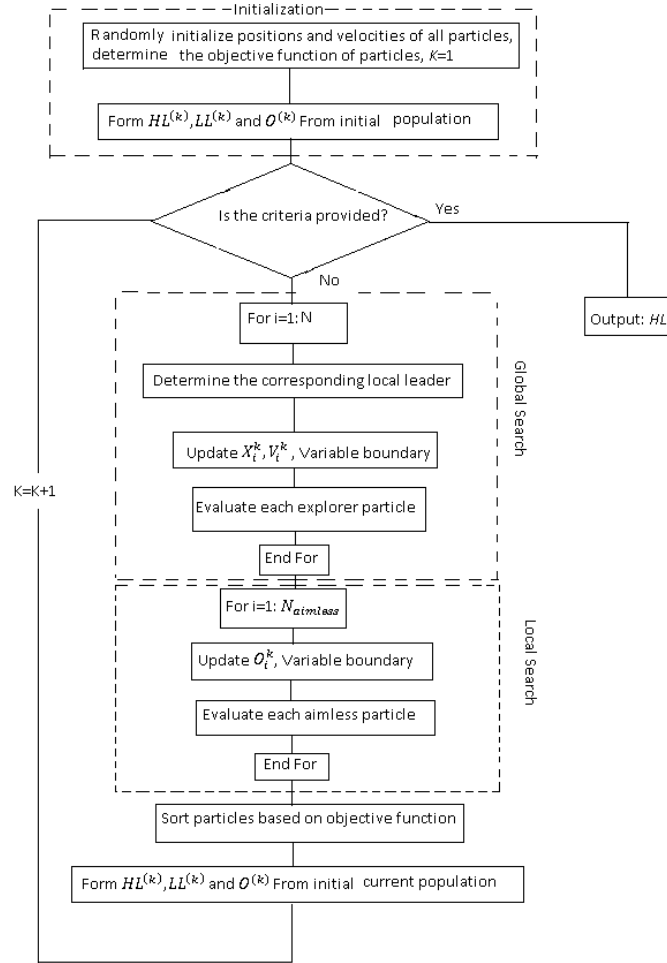


Figure 2. Flowchart of the HPSSO algorithm [14]

4. Simulation Results

4.1. Fitness Function

The fitness function provides the most appropriate solution for multi-objective optimization. The fitness function is given in Eq. 11. In this equation, P_L represents the sizing of distributed energy sources, LPSP is power supply loss, and RF is an environmental factor. The upper and lower limit of the problem are given in Eq. 12.

$$A = \frac{1}{(0.6*LPSP)+(0.2*RF)+(0.2*P_L)} \quad (11)$$

$$\begin{cases} 0 < P_{solar\ power\ plant} \leq 8000 \\ 0 < P_{wind\ turbine} \leq 4500 \\ 0 < P_{diesel\ generator} \leq 1500 \end{cases} \quad (12)$$

4.2. Comparison of SSO and HPSSO in terms of iteration

SSO and HPSSO algorithms were executed with different number of iterations as 100, 250, and 400 in three different cases. Each algorithm was executed 30 times and the mean values were taken. The fitness value graph for SSO algorithm with 100 iterations has been demonstrated in Fig. 2. The power values of the microgrid, renewable and non-renewable energy sources, the battery and the residual load to be supplied

are shown in Table 1. These initial values are values that algorithms do not optimize. The algorithm parameter is $n_{population}=100$.

Table 1. The initial power values

Problem parameters	Values
$P_{residence\ load}$	1000 kW
$P_{battery}$	4000 kW
$P_{battery\ load}$	3600 kW
$P_{solar\ power\ plant}$	8000 kW
$P_{wind\ turbine}$	4500 kW
$P_{diesel\ generator}$	1500 kW

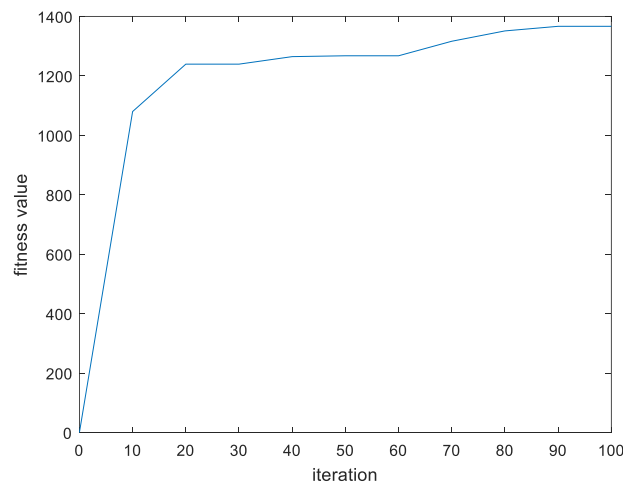


Figure 3. The fitness values obtained from the SSO algorithm in case of maximum 100 iterations

When the SSO algorithm is executed with 100 iterations, the obtained optimized power values for the solar power plant, the wind turbine, and the diesel generator are 4708.24 kW, 2355.65 kW, and 3076.029 kW respectively.

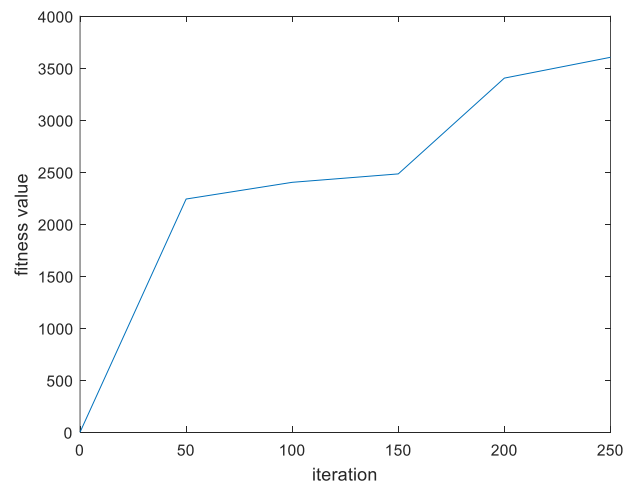


Figure 4. The fitness value graph for the SSO algorithm when the maximum number of iterations equals to 250

The obtained power values for the solar power plant, the wind turbine, and the diesel generator are 4625.96 kW, 2587.776 kW, and 3162.65 kW respectively after 250 iterations of SSO. In Fig. 3, the fitness values versus iteration numbers are shown.

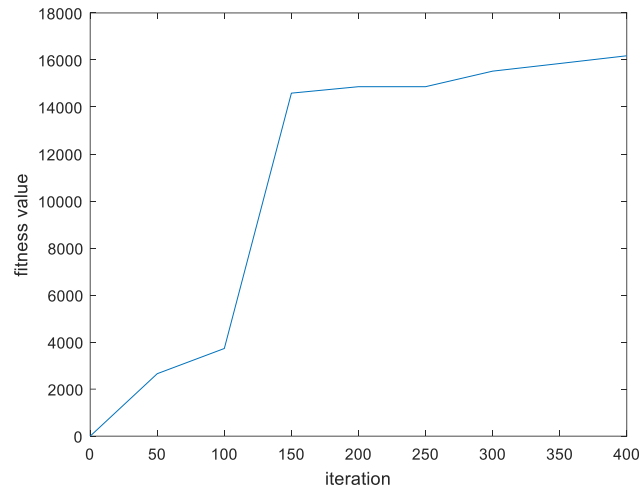


Figure 5. The fitness value for $n_{iteration} = 400$ in the SSO algorithm

When the SSO algorithm is executed with $n_{iteration} = 400$; the power values of the solar power plant, the wind turbine, and the diesel generator are 4988.17 kW, 2807.22 kW, 2692.87 kW. In Fig. 4, the fitness value versus the number of iterations is shown for $n_{iteration} = 400$.

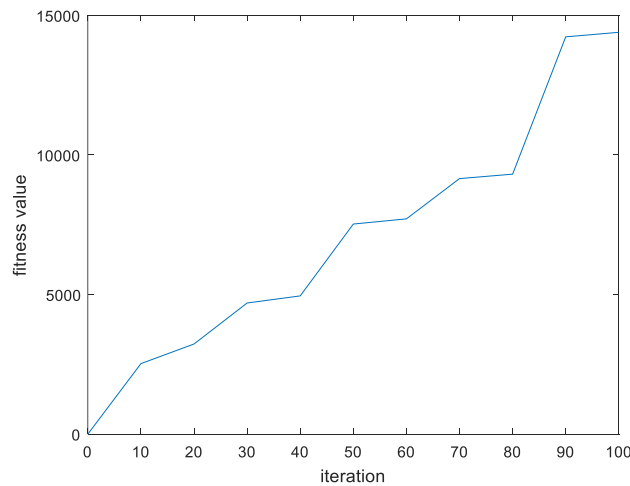


Figure 6. The fitness value for $n_{iteration} = 100$ in the HPSSO algorithm

The obtained optimized power values for the solar power plant, the wind turbine, and the diesel generator are 3594 kW, 2294.029 kW, and 3826.79 kW after 100 iterations of HPSSO. In Fig. 5, the fitness value and the number of iterations is shown for $n_{iteration} = 100$.

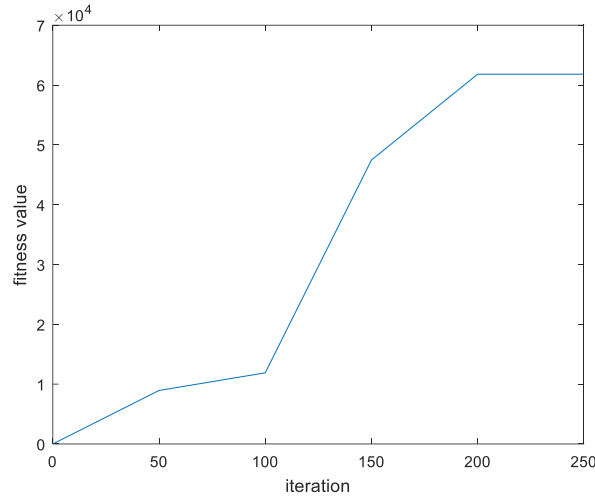


Figure 7. The fitness value graph for the HPSSO algorithm when the maximum number of iterations equal to 250

When the HPSSO algorithm is run with $n_{iteration} = 250$, the optimized power values for the solar power plant is the wind turbine is and the diesel generator is 3776.21 kW, 2462.21 kW, and 3675.81 kW respectively. In Fig. 6, the fitness value versus the number of iterations is demonstrated for the HPSSO algorithm.

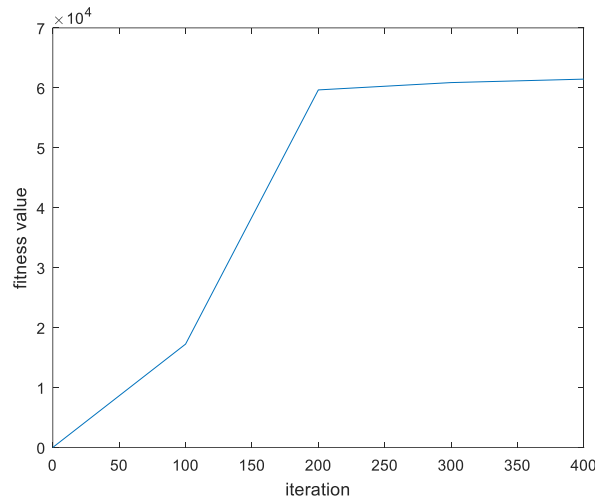


Figure 8. The fitness value for $n_{iteration} = 400$ in the HPSSO algorithm

The values of 3219.8 kW, 2173.37 kW, and 4253.42 kW have been obtained from the optimized power values for the solar power plant, the wind turbine, and the diesel generator respectively when the HPSSO algorithm is executed with 400 iterations. In Fig. 7, the fitness value versus the number of iterations is demonstrated for the HPSSO algorithm. Power values for different iterations of the SSO and HPSSO algorithms are given in Table 2. As seen in Table 2, as the number of iterations is increased in SSO algorithm, the power values of the solar power plant and wind turbine are increased. The best results in the HPSSO algorithm are achievable for $n_{iteration} = 60$.

Table 2. Comparison of iteration values of SSO and HPSSO

	Solar Power Plant	Wind Turbine	Diesel Generator
$n_{iteration_{SSO}} = 100$	4708.24 kW	2355.65 kW	3076.029 kW
$n_{iteration_{HPSSO}} = 100$	3594 kW	2294.029 kW	3826.79 kW
$n_{iteration_{SSO}} = 250$	4625.96 kW	2587.776 kW	3162.65 kW
$n_{iteration_{HPSSO}} = 250$	3776.21 kW	2464.21 kW	3675.81 kW
$n_{iteration_{SSO}} = 400$	4988.17 kW	2808.22 kW	2692.87 kW
$n_{iteration_{HPSSO}} = 400$	3219.8 kW	2173.37 kW	4253.42 kW

4.3. Comparison of SSO and HPSSO in terms of population

Three different population number values were selected as 20, 60, and 100 for the SSO and HPSSO algorithms. Each algorithm was run 30 times and mean values were taken. Table 1 presents the problem parameters. The maximum number of iterations is selected as 400.

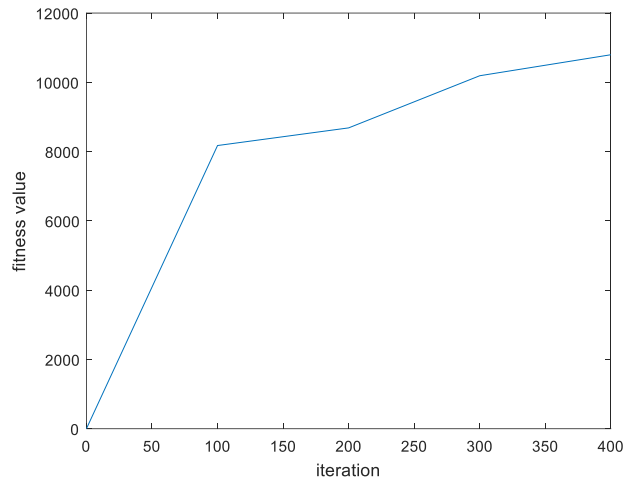


Figure 9. The fitness value for $n_{population} = 20$ in the SSO algorithm

When the number of population of SSO algorithm is selected as 20; the obtained power values for the solar power plant, the wind turbine, and the diesel generator are 5948.833 kW, 2502.78 kW, and 2218.05 kW respectively. Fig. 8 shows the fitness value versus the iterations for the SSO algorithm.

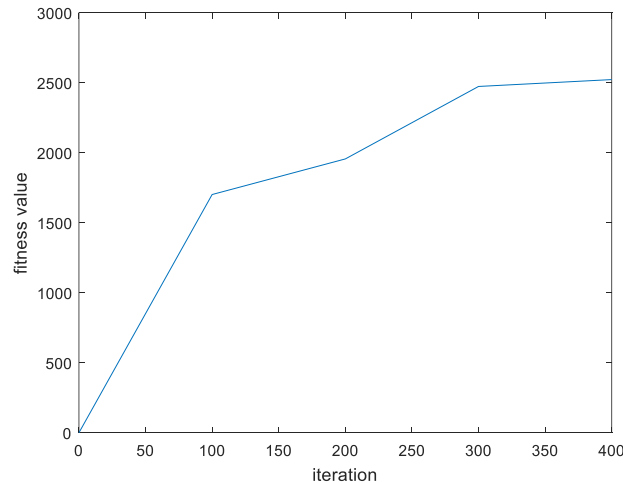


Figure 10. The fitness value changes for a maximum number of population 60 in the

SSO algorithm

The values of 6193.61 kW, 2761.58 kW, and 1861.651 kW have been obtained for optimized power values for the solar power plant, the wind turbine, and the diesel generator with SSO algorithm using 60 search agents. Fig. 9 shows the fitness value and the number of iterations for $n_{population} = 60$.

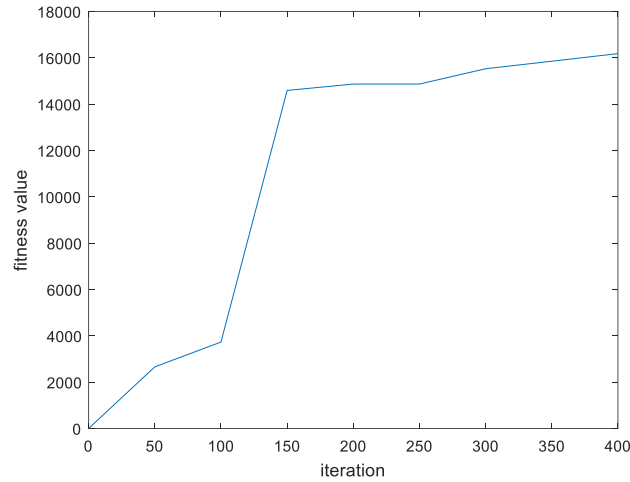


Figure 11. The fitness value change with 100 individuals in the SSO algorithm

When the SSO algorithm is executed with $n_{population} = 100$; the values of 4988.17 kW, 2807.22 kW, 2692.87 kW have been obtained for the power values of the solar power plant, the wind turbine, and the diesel generator respectively. Fig. 10 shows the fitness value and the number of iterations for $n_{population} = 100$.

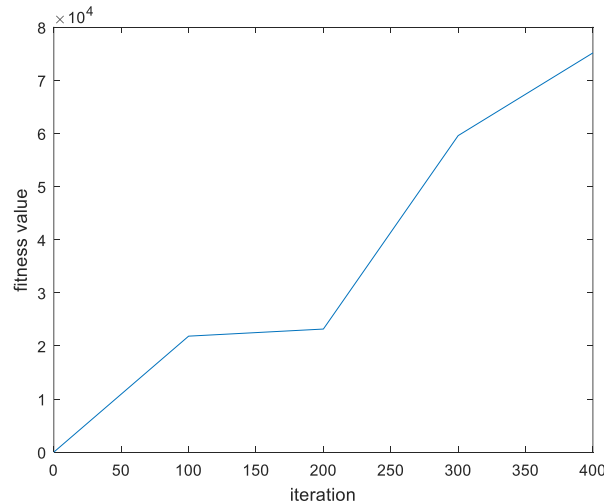


Figure 12. The fitness values change with 20 individuals in the HPSSO algorithm

When the HPSSO algorithm is run with 20 individuals; the optimized power values for the solar power plant, the wind turbine, and the diesel generator are 2649 kW, 1959.511 kW, and 4947.75 kW are obtained respectively. Fig. 11 shows the fitness value and the number of iterations for $n_{population} = 20$.

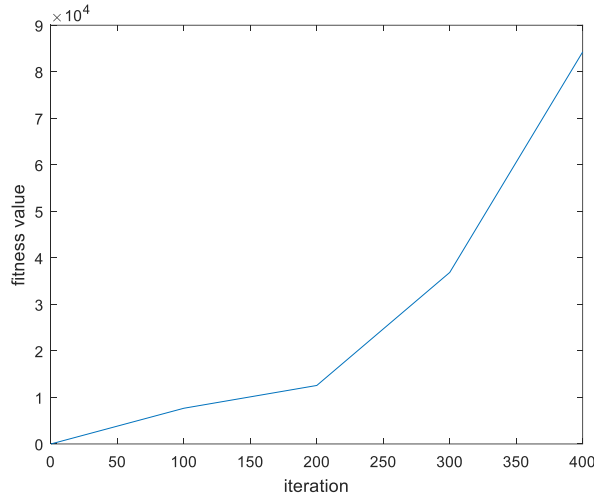


Figure 13. The fitness values change in the HPSSO algorithm with 60 individuals

When the HPSSO algorithm is run with 60 individuals; the values of 2577.43 kW, 1641.04 kW, and 5375.14 kW have been obtained for the optimized power values of the solar power plant, the wind turbine, and the diesel generator. Fig. 12 shows the fitness value and the number of iterations for $n_{population} = 60$.

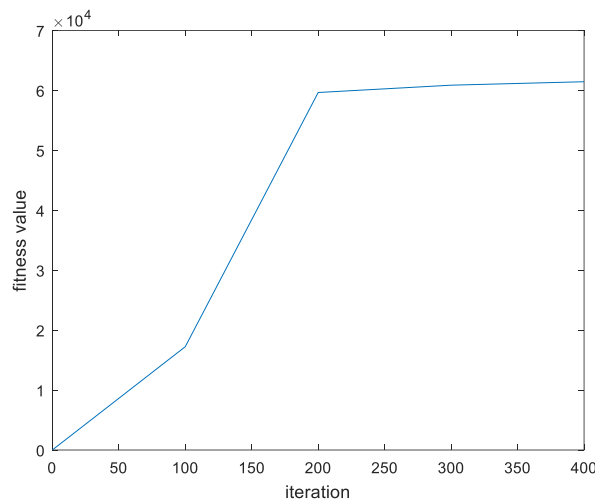


Figure 14. The fitness value change with 100 individuals in the HPSSO algorithm

When the HPSSO algorithm is executed with 100 individuals; the values of the power values for the solar power plant, the wind turbine, and the diesel generator are 3219.8 kW, 2173.37 kW, and 4253.42 kW are obtained. Fig. 13 shows the fitness value and the number of iterations for $n_{population} = 100$.

The power values for the different population size of the SSO and HPSSO algorithm are given in Table 3. The HPSSO algorithm yields better results as the population size increases. The SSO algorithm achieved good results for all population values However, this algorithm has given the best results for $n_{population} = 60$.

Table 3. Comparison of population values of SSO and HPSSO

	Solar Power Plant	Wind Turbine	Diesel Generator
$n_{population\ SSO} = 20$	5948.83kW	2502.78kW	2218.05kW
$n_{population\ HPSSO} = 20$	2649kW	1959.51kW	4947.75kW
$n_{population\ SSO} = 60$	6193.61kW	2761.58kW	1861.651kW
$n_{population\ HPSSO} = 60$	2577.43kW	1641.04kW	5375.14kW
$n_{population\ SSO} = 100$	4988.17kW	2807.22kW	2692.87kW
$n_{population\ HPSSO} = 100$	3219.8kW	2173.37kW	4253.42kW

5. Conclusions

In this study two new metaheuristic algorithms, namely SSO and HPSSO, have been designed as solution search methods for the problem reducing the emission, increasing reliability, and optimizing the sizing of the microgrid for the first time. A microgrid consisting of 8 MW solar panel, 4.5 MW wind turbine, 15 MW diesel generator, and 4 MW battery has been taken into consideration. The microgrid feeds a 10MW network. In Turkey, the inductive reactive penalty limit is 33% for facilities with a connection power of less than 50KVA. In our system, there is a residence as a load and this ratio is not exceeded and only active power is taken into consideration and all calculations are made accordingly.

The results have been obtained by running SSO and HPSSO algorithms on three different iterations as 100, 250, and 400. The SSO algorithm has given the best solution with 400 iterations. The best results of 4988.17 kW, 2808.22 kW, and 2692.87 kW for the solar panel, wind turbine, and diesel generator respectively have been achieved. The best results have been obtained from the HPSSO algorithm for 250 iterations. These obtained results for the solar panel, wind turbine, and diesel generator are 3776.21 kW, 2464.21 kW, and 3675.81 kW respectively. Comparing SSO and HPSSO in terms of different iteration values, SSO algorithm has been found to optimize energy resources better.

The powers of the solar power plant, wind turbine, and diesel generator are also aimed to be optimized by the SSO and the HPSSO with three different population sizes like 20, 60, and 100. The SSO algorithm obtained optimal results when the population size has been set at 60. These results are 6193.61 kW, 2761.58 kW and 1861.651 kW for the solar power plant, wind turbine, and diesel generator respectively. In the HPSSO algorithm, when the number of population is increased, and the yield is also increased. This algorithm has found the results of 3219.8 kW, 2173.37 kW, and 4253.42 kW for the solar power plant, wind turbine, and diesel generator using 100 individuals. The SSO algorithm yielded better results with less population when compared with HPSSO. This also simplifies the process and shortens the process time.

Novel and efficient metaheuristic algorithms are still being proposed for efficient results for many types of problems. Their new, adaptive, hybrid versions with optimized parameters may be used for the optimizing of the microgrids. Furthermore, this type of optimization, in fact, contains many objectives and thus efficient real multi-objective and many-objective versions of these metaheuristic algorithms may also be designed for these types of problems. SSO and HPSSO are two new metaheuristic algorithms. These algorithms have been applied by using the different parameters. Both algorithms have shown very good significant improvement. These algorithms can improve the solution quality of many complex problems. They can also be applied to different engineering problems.

References

- [1] Wang, Z., *et al.*, Intelligent Multi-Agent Control for Integrated Building and Micro-Grid Systems. *In Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES, 2011*, pp.1-7.



- [2] Aziza, M., & Walling, F., Multi-objective Particle Swarm Optimization of Hybrid Micro-Grid System: A Case Study in Sweden, *Energy*, 2017, Vol.123, pp. 108-118.
- [3] Fossati, J. P., *et al.*, A Method for Optimal Sizing Energy Storage Systems for Microgrids. *Renewable Energy*, 2015, Vol.77, pp.539-549.
- [4] Bevrani, H., *et al.*, Intelligent Frequency Control in an AC Microgrid: Online PSO-Based Fuzzy Tuning Approach. *IEEE Transactions on Smart Grid*, 2012, Vol. 3(4), 1935-1944.
- [5] Chen, Y. K., *et al.*, Design and Implementation of Energy Management System with Fuzzy Control for DC Microgrid Systems. *IEEE Transactions on Power Electronics*, 2013, Vol. 28(4), 1563-1570.
- [6] Seung, C., *et al.*, Agent based Particle Swarm Optimization for Load Frequency Control of Distribution Grid, *In Universities Power Engineering Conference (UPEC), 2012 47th International*, 2012, pp. 1-6.
- [7] Eilaghi, S. F., *et al.*, Optimal Voltage Unbalance Compensation in a Microgrid Using PSO Algorithm. *In Power India International Conference (PIICON), 2016 IEEE 7th*, 2016, pp.1-6
- [8] Yazdani, M., & Jolai, F., Lion Optimization Algorithm (LOA): A Nature-inspired Metaheuristic algorithm. *Journal of Computational Design and Engineering*, 2016, Vol. 3(1): pp.24-36.
- [9] Gandomi, A. H., *et al.*, Mixed Variable Structural Optimization Using Firefly Algorithm, *Computers & Structures*, 2011, Vol. 89(23-24), pp. 2325-2336.
- [10] Gandomi, A. H., & Alavi, A. H., Krill Herd: A New Bio-inspired Optimization Algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 2012, Vol. 17(12), pp. 4831-4845.
- [11] Revathi, K., & Krishnamoorthy, N., The Performance Analysis of Swallow Swarm Optimization Algorithm, *2015 2nd International Conference on IEEE*, 2015, pp. 558-562.
- [12] Bouzidi, S., & Riffi, M. E., Discrete Swallow Swarm Optimization Algorithm for Travelling Salesman Problem, *In Proceedings of the 2017 International Conference on Smart Digital Environment*, 2017, pp. 80-84.
- [13] Neshat, M., *et al.*, Swallow Swarm Optimization Algorithm: A New Method to Optimization, *Neural Computing and Applications*, 2013, Vol. 23(2), pp.429-454.
- [14] Kaveh, A., *et al.*, Hybrid PSO and SSO Algorithm for Truss Layout and Size Optimization Considering Dynamic Constraints, *Structural Engineering and Mechanics*, 2015, Vol. 54(3), pp. 453-474.
- [15] Sam C., & Ali, A., A., 2015. Identification of Crack in a Cantilever Beam using Improved PSO Algorithm, *International Journal for Innovative Research in Science & Technology*, 2015, Vol.1(11), pp. 454- 461.