ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20200112201457
Original scientific paper

Multi-Objective Optimization of Microgrids Based on Recent Metaheuristic Methods

Tuba TANYILDIZI AĞIR*, Zafer AYDOĞMUŞ, Bilal ALATAŞ

Abstract: As the technology develops in the modern world, the need for electrical energy has increased. Renewable energy sources have emerged as an alternative energy source to fossil energy sources. Micro grids are the hybrid energy sources for both renewable and non-renewable energy sources. The choice of the microgrid depends on meeting the supply and low cost requirements while avoiding environmental pollution. Therefore, emission, reliability and sizing of a micro grid have been investigated in the present study. In addition, Swallow Swarm Optimization (SSO) and Hybrid Particle Swallow Swarm Optimization (HPSSO) algorithms were not found in micro grid related optimization studies. Performance of SSO and HPSSO algorithms was also evaluated. Particle Swarm Optimization (PSO), SSO, and HPSSO were adjusted in this study as multi-objective optimization method for increasing the reliability, decreasing emission and sizing energy resources of a microgrid feeding a 10 MW residence. 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 efficiencies of these algorithms were compared for different iterations and populations. In this study, the best results were obtained with the SSO algorithm. Loss of power supply probability (*LPSP*) = 0, Renewable factor (*RF*) = 1, with this algorithm our micro-grid has achieved a safe energy and minimum emission to feed the residence. In addition, a system that connects and disconnects the energy resources in varying load conditions was actualized with the SSO algorithm. With this algorithm *LPSP* = 0, *RF* = 1, P_{size} = 0,001. Maximum reliability, zero emission and minimum sizing of the energy sources in our microgrid were achieved with loads of up to 50%. Moreover, *LPSP* = 0.39, *RF* = 0.086, P_{size} = 0,21 values were obtained for loads 50% and above and good results were obtained for reliability, emission and sizing of energy sources.

Keywords: hybrid particle swallow swarm optimization; meta heuristic algorithms; microgrid; particle swarm optimization; swallow swarm optimization

1 INTRODUCTION

In comparison with the traditional and centralized power generation systems, future generation systems are small-scale and non-centralized grids comprised of Renewable Energy Resources (RES). Moreover, power quality and operation stability may be affected adversely due to the intermittent and uncertain characteristics of RES. Distribution generation and microgrid technology provide a possible solution to balance RES [1]. The concept of microgrid has come to the forefront due to increased energy demand, the operation of the grid close to stability limits, costly transmission infrastructure and the increasing impact of Distributed Energy Resources (DER). Microgrids are small power distribution comprised of DER, battery storage systems and loads [2]. Microgrid provides a competitive contribution to market models and new technologies [3]. There is an increasing interest to microgrid in many countries due to its ability to meet user demands, its restructuring of the electrical industry as well as its advantages for transmission and distribution systems. Microgrids are operated in two modes: grid-connected and islanded. Under normal conditions, a microgrid is connected in parallel to the grid and operated as such. Some grid-connected systems comprise a backup system to be used when RES is not sufficient to feed the microgrid [4]. Optimization is the action of acquiring the best result under certain limitations and conditions. In practice, optimization is used to determine the minimum value of a function. Mathematical programming techniques are generally applied for determining the minimum value of the variables generally under certain limitations. The formula is given in Eq. (1):

$$\text{find } X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ x_n \end{pmatrix} \text{which minimizes } f(X) \tag{1}$$

Subject to the constraints:

$$g_j \leq 0, j = 1, 2, \ldots, m \tag{2}$$

$$l_j = 0, j = 1, 2, \ldots, p \tag{3}$$

where, $f(X)$ is the objective function. X is an n-dimensional design vector. $g_j(x)$ and $l_j(x)$ are inequality and equality constraints. n is the number of variables while m represents the number of inequality constraints. There are many different optimization techniques available subject to the characteristics of the problem. There may be more than one solution for any design problem. Hence, an objective function should be selected for comparing the alternative acceptable solutions in order to select the best solution. The selection of the objective function depends on the problem. Some objectives conflict with each other. The structuring of the objective function makes up the most important decisions for the whole optimum design process. The objective functions can be examined under four categories: technical, economical, environmental, and socio-political [5]. Metaheuristic algorithms have displayed a promising performance for solving the non-linear and real-life optimization problems. All metaheuristic algorithms display a global search and random variation. These algorithms may provide a good solution for difficult optimization problems in a reasonable amount of time. Metaheuristic algorithms are used effectively in many fields for global optimization [6].

PSO is a metaheuristic algorithm inspired by birds. PSO starts a population of particles that registers solutions and looks for the next best solution. PSO explores the variable area depending on the fitness function of the best solution [7]. Characterized by a position vector and a velocity vector in the optimization process, each particle benefits from its own search trajectory and the historical experience of its neighbor. Since PSO can be applied easily in practice, it has been used in the solution of real-world optimization problems [8]. Neshat et al. proposed the SSO algorithm, which is an optimization technique based on herd intelligence. Swallows have high intelligence at very high speed. These birds communicate by making different sounds in different situations [9]. SSO algorithm has been used in many applications, inspired by the intelligent social life relationship of swallows such as high-speed flight, hunting skills and migration of large groups [10]. The HPSSO algorithm was created by including the most important features of SSO in the PSO formulation. The HPSSO algorithm has a mechanism that allows them to learn not only the most experienced particles, but also promising particles [11].

Azaza and Wallin investigated the potential in different Swedish cities for an independent hybrid microgrid investment. The multiobjective PSO has been used to find the optimal sizing for each location.The results showed that the hybrid microgrid has significant potential in Sweden [12]. Bukar et al. used Grasshopper optimization algorithm (GOA), PSO, cuckoo search optimization algorithm to size the microgrid in the Yobe State of Nigeria. The simulation results showed that GOA has the most appropriate size compared to other algorithms and the system cost is reduced by 14% [13]. Kerdphol et al. sized the battery to avoid micro grid instability and system crash after power failure. They proposed an optimum battery size using frequency control based on the PSO method. They conclude that this method can significantly increase the stability of the power system, network security and planning flexibility [14]. Trivedi et al used Whale Optimization algorithm (WOA), Ant Colony Optimization, Gradient Method and PSO to minimize the fuel cost and emission of the microgrid. Compared to other algorithms, WOA achieved the lowest cost [15]. Wang et al. created a moth flame optimization model to obtain the minimum operating cost of the microgrid. The results showed that the optimization model can reduce the cost effectively [16]. Javid et al. used the Hybrid Optimization Model for Electric Renewables method to reduce the cost and emission of the microgrid. The results showed that the proposed model reduced the cost and emission of the microgrid [17]. Bouzıdı and Rıffı benefited from SSO to solve the famous problem of the traveling salesman. The experimental results on the selected samples showed that SSO is more effective when compared to other metaheuristic methods [18]. In Revathi and Krishmoorthy Soft computing, they used PSO, artificial fish swarm optimizastion and SSO algorithms. The SSO algorithm has proven to have a faster convergence rate with a lower number of iterations [19]. Neshat et al. benefited from the PSO, Artifical Fish Swarm Optimization and SSO algorithm for the simultaneous calculation of multimodal functions. The SSO algorithm has been tested with the benchmark function 19. The SSO algorithm has optimized the problem better than other optimization methods [20]. Kaveh et al. tested the SSO and HPSSO algorithms by solving eleven mathematical optimization problems and six truss weight minimization problems. The numerical results prove that the HPSSO algorithm performs better than the standard PSO and PSO enhanced versions. The HPSSO algorithm also provided a good balance between global and local search [11]. C. Sam. and Ali used the PSO and HPSSO algorithm to estimate the location and depth of a crack in the krish structure. The HPSSO algorithm has shown very good accuracy in the crack definition with less calculation effort [21]. Kaveh et al. proposed the PSO, SSO and HPSSO algorithm for optimization of highly nonlinear dynamic truss shape and size optimization with multiple natural frequency. The HPSSO algorithm displayed a higher performance compared to other optimization techniques [22].

The operation of the microgrid at maximum performance is not a single purpose. The efficient operation of the microgrid depends on many reasons such as being able to meet the energy demand, being a clean energy and having minimum sizing. If these goals are optimized, the performance of the micro-grid increases and its cost decreases. In this study, emission, sizing and reliability of the microgrid have been optimized with the help of PSO, SSO and HPSSO algorithms.

SSO and HPSSO algorithms have been applied in different fields of science and successful results have been obtained. However, these optimization methods have never been used before to optimize the micro grid. Therefore, a study has been conducted on SSO and HPSSO in order to add innovation to science. Simulation results showed that SSO and HPSSO algorithms are suitable for microgrid. Contrary to majority of the studies in which HPSSO algorithm displayed a better performance, SSO delivered a better performance in this study.

The aim of this study was to increase the reliability of the micro grid and to minimize emission and sizing. For this purpose, PSO, SSO and HPSSO algorithms were used. The PSO algorithm has been used in many scientific studies to optimize the microgrid and successful results have been obtained. This algorithm is used to compare the new optimization methods of the SSO and HPSSO algorithms. If successful results are obtained with the SSO and HPSSO algorithms, it can be applied to different studies related to micro grid. Optimization can be improved or the objectives in the microgrid can be changed if poor results are achieved. In addition, these optimization techniques can contribute to different areas of science.

All optimization methods used in the study were compared via tables and graphs. The study was organized as follows: the Sect. 2 covers the theoretical background, Sect. 3 describes the suggested method, Sect. 4 presents the optimization results and Sect. 5 presents the results.

2 THEORETICAL BACKGROUND

Engineering design problems generally contain complex objective functions with a large number of decision variables. A long period of time is required for solving engineering design problems via traditional search methods. Even though these methods yield promising results for many problems, they may fail in more complex design problems. There may be a large number of decision variables in actual design problems and their impacts on the objective function may be too complex. More efficient optimization methods are needed in such cases. Metaheuristic algorithms displayed a promising performance for the solution of non-linear real-world optimization problems. All metaheuristic algorithms use a certain random change and perform a global search. These algorithms may provide optimal or near-optimal good solutions for difficult optimization problems in a reasonable time [23].

There are many reasons for the popularity and success of metaheuristics. These algorithms were developed by imitating the most successful biological, physical, and chemical processes in real life. The best performance of metaheuristics in nature is based on how well they imitate the best characteristic of nature. It is a common criterion that the best remains alive. The current best solution is updated over and over again. Moreover, certain elitism is required. This should indicate that the best or the most proper solutions are not lost and are passed down to posterity [24].

Even though metaheuristic algorithms try to reach their goals subject to their paradigms and philosophies, some of these are specialized for certain cases. This has increased the need for hybrid metaheuristics. Hybrid metaheuristics is an algorithm that does not follow a single traditional metaheuristic concept. Moreover, it combines various algorithmic ideas outside the field of metaheuristic. The hybridization of different algorithmic concepts generally displays a better performance by combining the advantages of different algorithms [25].

The common properties of almost all metaheuristic algorithms can be stated as follows: they get inspiration from nature, they use random variables, they do not require significant gradient knowledge and the problem has several parameters. Metaheuristic algorithms have many advantages with regard to robustness and performance for noise environments. High performance metaheuristics are required for overcoming certain optimization problems. These are some of the metaheuristic algorithms: Genetic algorithm, PSO, differential evolution, artificial bee colony algorithm, SSO, HPSSO, cuckoo algorithm, ant colony optimization, simulated annealing, artificial immune systems, and tabu search [27].

2.1 Particle Swarm Optimization

PSO is an optimization method with inspiration from nature and has been introduced by Kneddy and Eberhart. The PSO algorithm that imitates the social behaviours of bird and fish swarms cooperates and interacts to search for a solution in the problem space. All particles in PSO continue learning from the best personal experiences p_{best} and the best global solution of the swarm (g_{gest}) [27].

PSO is a stochastic algorithm and does not require gradient knowledge for its objective function. The PSO method may provide high quality solutions at shorter calculation times in comparison with other stochastic methods [28]. When compared with other evolutionary algorithms, PSO does not have many parameters. PSO has a concise framework that can easily be implemented [29].

The particles in the algorithm initially start a search space with the following characteristics: (i) position and velocity (ii) objective function of these positions (iii) neighbouring information: the best position and objective function value (iiii) storage of the best position found. The stopping criteria are determined by two conditions: stopping at a predetermined number of repetitions or when it is not possible to make any improvement [31].

ω denotes the inertia weight, while r_1 and r_2 is the vector of random values between 0 and 1, c_1 and c_2 are cognitive and social components, p_{best} is the best position of the particle g_{best}, is the best position of the swarm, v is the particle velocity, x is the particle position. The particle velocity is calculated via Eq. (4) and the particle position is calculated via Eq. (5) [31].

$$v_{i+1} = \omega v_{\mathrm{i}} + c_1 r_1 (p_{best} - e_i) + c_2 r_2 (g_{best} - e_i) \tag{4}$$

$$x_{i+1} = x_{\mathrm{i}} + v_{i+1} \tag{5}$$

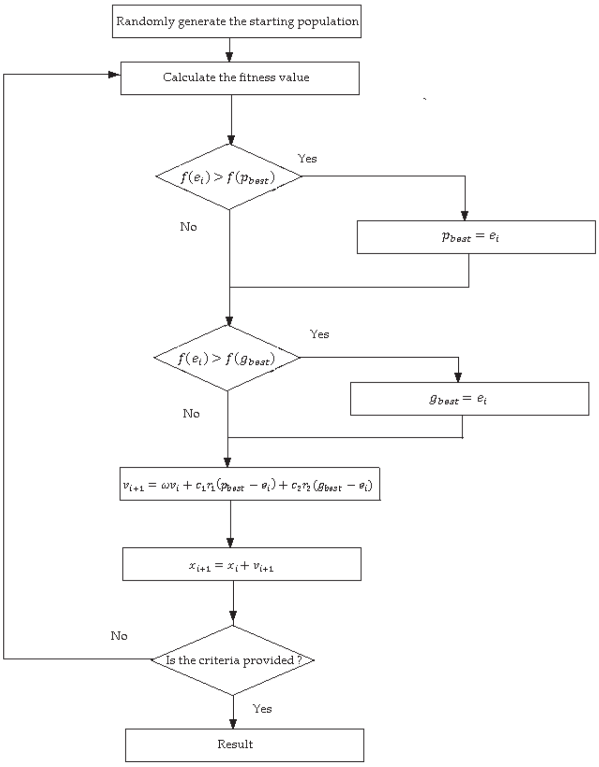
Randomly generate the starting population
Calculate the fitness value
$f(e_i) > f(p_{best})$
Yes
No
$p_{best} = e_i$
$f(e_i) > f(g_{best})$
Yes
No
$g_{best} = e_i$
$v_{i+1} = \omega v_i + c_1 r_1 (p_{best} - e_i) + c_2 r_2 (g_{best} - e_i)$
$x_{i+1} = x_i + v_{i+1}$
No
Is the criteria provided ?
Yes
Result

Figure 1 PSO flow diagram

The starting population is started randomly as seen in the flowchart shown in Fig. 1. The fitness values of all particles in the population are calculated. If the value of e_i is greater than p_{best}, it is equal to p_{best}, while the value of e_i particle is equal to itself if its value is not greater than p_{best}. If the value of e_i particle is greater than g_{best}, it is equal to gbest, while e_i is equal to itself if its value is not greater than g_{best}. The velocity and position values of all particles in the population are calculated. If the criterion is met, the iteration is stopped and if the criterion is not met, the iteration returns to the beginning and the fitness value of the particles is calculated.

2.2 Swallow Swarm Optimization

Neshat et al. proposed SSO based on the collective movement of swallows [18]. Each colony is divided into sub-colonies with neighbouring positions in a region. Each colony generally has an experienced leader. If the suitability of the group for leadership decreases for some reason, another bird with higher suitability takes its place. Swallows always follow the leader on the condition that it has the required abilities. There are two types of leaders: first is the local leader that manages the internal colony and points out a local solution. The second is the head leader which is the particle that is responsible from the leadership of the whole colony and points at the best solution point

[20]. The explorer particle is responsible from searching the problem space. This search behaviour takes place under the impact of a series of parameters [32].

- Position of the local leaders.
- Position of the global leaders.
- Best individual experience along the path.
- Previous path.

Each particle has a personal feature. However, their common feature is having a flying central colony. Every particle examines the environment with an adaptive radius and displays smart behaviours. The states of the neighbouring particles are accepted as global leader and local leader. These particles move parallel to each other and they always interact. Actual borders between the sub-colonies are never marked since swallows move at very high velocities and dynamics. Advantages of the SSO algorithm can be listed as below [33]:

- Move faster on flat surfaces.
- Display smart contribution in different groups.
- Have high convergence rates.
- Acquire local extremum points.

$$\left\{\begin{array}{l} \alpha_{LL} = \{\text{if}\left(e_i = 0 \| e_{best} = 0 \rightarrow 2\right. \\ \alpha_{LL} = \text{if}\left(e_i < e_{best}\right) \&\& \left(e_i < LL_i\right) \rightarrow \left.\frac{rande_i}{e_i e_{best}}\right) e_i,\ e_{best} \neq 0 \\ \alpha_{LL} = \text{if}\left(e_i < e_{best}\right) \&\& \left(e_i > LL_i\right) \rightarrow \left.\frac{2rande_{best}}{1/(2e_i)}\right) e_i \neq 0 \\ \alpha_{LL} = \text{if}\left(e_i > e_{best}\right) \rightarrow \left.\left.\frac{e_{best}}{1/(2 \times rand)}\right)\right) \end{array}\right\} \tag{6}$$

$$\left\{\begin{array}{l} \beta_{LL} = \{\text{if}\left(e_i = 0 \| e_{best} = 0 \rightarrow 2\right. \\ \beta_{LL} = \text{if}\left(e_i < e_{best}\right) \&\& \left(e_i < LL_i\right) \rightarrow \left.\frac{rand()e_i}{e_i LL_i}\right) e_i,\ LL_i \neq 0 \\ \beta_{LL} = \text{if}\left(e_i < e_{best}\right) \&\& \left(e_i > LL_i\right) \rightarrow \left.\frac{2rand()LL_i}{1/(2e_i)}\right) e_i \neq 0 \\ \beta_{LL} = \text{if}\left(e_i > e_{best}\right) \rightarrow \left.\left.\frac{LL_i}{1/(2 \times rand)}\right)\right) \end{array}\right\} \tag{7}$$

$$V_{LL_{i+1}} = V_{LL_i} + \alpha_{LL} rand\left(e_{best} - e_i\right) + \beta_{LL} rand\left(LL_i - e_i\right) \tag{8}$$

$$\left\{\begin{array}{l} \alpha_{HL} = \{\text{if}\left(e_i = 0 \| e_{best} = 0 \rightarrow 1.5\right. \\ \alpha_{HL} = \text{if}\left(e_i < e_{best}\right) \&\& \left(\left(e_i\right) < HL_i\right) \rightarrow \left.\frac{rand()e_i}{e_i \times e_{best}}\right| e_i,\ e_{best} \neq 0 \\ \alpha_{HL} = \text{if}\left(e_i < e_{best}\right) \&\& \left(\left(e_i\right) > HL_i\right) \rightarrow \frac{2 \times rand()e_{best}}{1/(2 \times e_i)} e_i \neq 0 \\ \alpha_{HL} = \text{if}\left(e_i > e_{best}\right) \rightarrow \left.\left.\frac{e_{best}}{1/(2 \times rand)}\right)\right) \end{array}\right\} \tag{9}$$

$$\left\{\begin{array}{l} \beta_{HL} = \{\text{if}\left(e_i = 0 \| e_{best} = 0\right) \rightarrow 1.5 \\ \beta_{HL} = \text{if}\left(e_i < e_{best}\right) \&\& \left(e_i < HL_i\right) \rightarrow \left.\frac{rand()e_i}{e_i \times HL_i}\right) e_i,\ \mathrm{H}L_i \neq 0 \\ \beta_{HL} = \text{if}\left(e_i < e_{best}\right) \&\& \left(e_i > HL_i\right) \rightarrow \left.\frac{2 \times rand()\mathrm{H}L_i}{1/(2 \times e_i)}\right) e_i \neq 0 \\ \beta_{LL} = \text{if}\left(e_i > e_{best}\right) \rightarrow \left.\left.\frac{HL_i}{1/(2 \times rand)}\right)\right) \end{array}\right\} \tag{10}$$

$$V_{HL_{i+1}} = V_{HL_i} + \alpha_{HL} rand\left(e_{best} - e_i\right) + \beta_{HL} rand\left(HL_i - e_i\right) \tag{11}$$

$$V_{i+1} = V_{HL_{i+1}} + V_{LL_{i+1}} \tag{12}$$

$$e_{i+1} = e_i + V_{i+1} \tag{13}$$

Eq. (8) is used to calculate the leader velocity, Eq. (11) is used to calculate the velocity of the lead leader, Eq. (12) is used to calculate the particle velocity and Eq. (13) is used to calculate the position of the particle [20].

$V_{HL_{i+1}}$ is the head leader velocity, α_{HL} is the acceleration coefficient of the head leader, rand 0 - 1 is a random number, e_{best} is the best value for the explorer particle, e_i is the explorer particle and β_{HL} is the acceleration coefficient for the head leader, HL_i is the head leader and V_{LL_i} is the velocity of the local leader, V_{i+1} is the velocity of the explorer particle, e_{i+1} is the explorer particle.

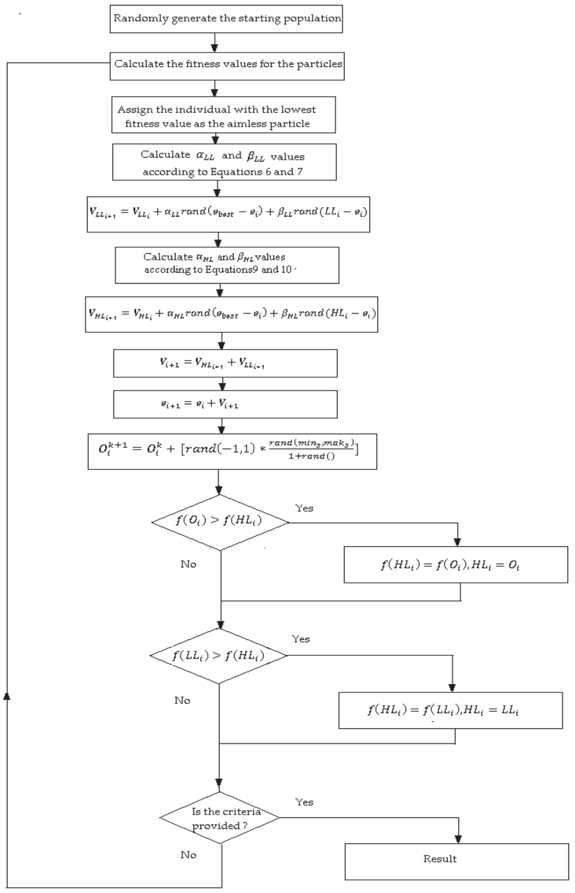
Randomly generate the starting population
Calculate the fitness values for the particles
Assign the individual with the lowest fitness value as the aimless particle
Calculate α_{LL} and β_{LL} values according to Equations 6 and 7
$V_{LL_{i+1}} = V_{LL_i} + \alpha_{LL} rand(e_{best} - e_i) + \beta_{LL} rand(LL_i - e_i)$
Calculate α_{HL} and β_{HL} values according to Equations9 and 10
$V_{HL_{i+1}} = V_{HL_i} + \alpha_{HL} rand(e_{best} - e_i) + \beta_{HL} rand(HL_i - e_i)$
$V_{i+1} = V_{HL_{i+1}} + V_{LL_{i+1}}$
$e_{i+1} = e_i + V_{i+1}$
$O_i^{k+1} = O_i^k + [rand(-1,1) * \frac{rand(min_s, mak_s)}{1+rand()}]$
$f(O_i) > f(HL_i)$
Yes
No
$f(HL_i) = f(O_i), HL_i = O_i$
$f(LL_i) > f(HL_i)$
Yes
No
$f(HL_i) = f(LL_i), HL_i = LL_i$
Is the criteria provided ?
Yes
No
Result

Figure 2 SSO flow diagram

The starting population is randomly generated as seen in the flowchart of Fig. 2. The fitness value of all individuals in the population is calculated. The particle with the smallest fitness value is determined as the aimless particle. Calculate α_{LL} and β_{LL} according to Eq. (6) and Eq. (7). Calculate the velocity of the local leader. Calculate α_{HL} and β_{HL} according to Eq. (9) and Eq. (10). Calculate the velocity of the head leader. Calculate the velocity and position of particles in the population. Calculate the aimless particles. If the aimless particle is greater than the head leader, the aimless particle is equal to the head leader, while the aimless particle is not greater than the head leader, the aimless particle is equal to itself. If the local leader is greater than the head leader, the local leader is equal to the head leader, while the local leader is equal to himself if the local leader is not greater than the head leader. If it meets the criteria, stop the iteration. If it does not meet the criteria, repeat the iteration.

2.3 Hybrid Particle Swallow Swarm Optimization

HPSSO algorithm is a hybrid algorithm comprised of PSO and SSO algorithms. HPSSO consists of explorer particle, aimless particle, and leaders. Explorer particle is the particle that carries out explorations in the design space. The aimless particle is the individual with the worst fitness value whereas leaders have the best fitness value in the colony. The initial population, velocity, and position are started randomly. The population velocity and position are updated until the iteration ends. When compared with PSO, it contains an additional term for explaining the contribution of the local leader [22]. Eq. (14) and Eq. (15) show the velocity and position formula respectively. Three possible options are taken into consideration in Eq. (16) for the objective particle. (i) only a random search is made as is the case in SSO. (ii) a local search is made in the neighboring of the local leaders. (iii) a dynamic search is made in the neighboring of global leaders.

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{14}$$

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 \left(P_i^k - X_i^k\right) + c_2 r_2 \left(P_g^k - X_g^k\right) + c_3 r_3 \left(P_{l(i)}^k - X_i^k\right) \tag{15}$$

$$O_i^{k+1} = \begin{cases} P_{l(i)}^k \text{scenario } 1 + \operatorname{rand}(-1,\ 1) \times \left[\left(\lambda^k \times \left(\max_s - \min_s\right)\right)\right] \\ P_{l(i)}^k \text{scenario } 2 + \operatorname{rand}(-1,\ 1) \times \left[\left(\lambda^k \times \left(\max_s - \min_s\right)\right)\right] \\ P_{l(i)}^k \text{scenario } 3 + \operatorname{rand}(-1,\ 1) \times \left[\left(\lambda^k \times \left(\max_s - \min_s\right)\right)\right] \end{cases} \tag{16}$$

$$\lambda^k = \lambda_{\max} - \left(\lambda_{\max} - \lambda_{\min}\right) \times \frac{iter}{iter_{\max}} \tag{17}$$

X_i^{k+1} is the updated position of the particle, X_i^k is the previous position of the particle, V_i^{k+1} is the updated velocity of the particle, ω is the weight of inertia that controls the impact o of previous velocity, while r_1, r_2, r_3 represent a random number in the interval (0, 1), c_1, c_2, c_3 is the learning factor, $P_{l(i)}^k$ is the local leader of the sub-colony that the ith particle contains. rand (−1, 1) is a random number between 1 and −1. $\min_s$ and $\max_s$ are the lower and upper limits of the design variables. λ^k is a parameter defined for creating an effective search interval for the local leader or global optimum. $\lambda_{\max}$ and $\lambda_{\min}$ are the first and last iteration values of the algorithm. In the present study, they were set as 0.01 and 0.001. Population update was as follows [11]. Fig. 3 shows the explanation for the flow diagram of the HPSSO algorithm.

- Copying the head leader and the local leader from one belt to the next can be interpreted as an elite strategy.
- Search carried out by the explorer particle for moving the population towards the best region of the design area.
- A dynamic search by the aimless particle in the neighboring of the local leader and head leader.

The starting population is selected randomly, as can be seen from the flowchart in Fig. 3. The fitness value of each particle is calculated. Calculate the global leader, local leader and aimless particle. Calculate the velocity and position of the particles. Calculate the fitness value of the explorer particle. Update the aimless particles according to Eq. (16). Calculate the fitness value of the particles. If the criterion is met, stop the iteration. If the criteria are not met, repeat the iteration.

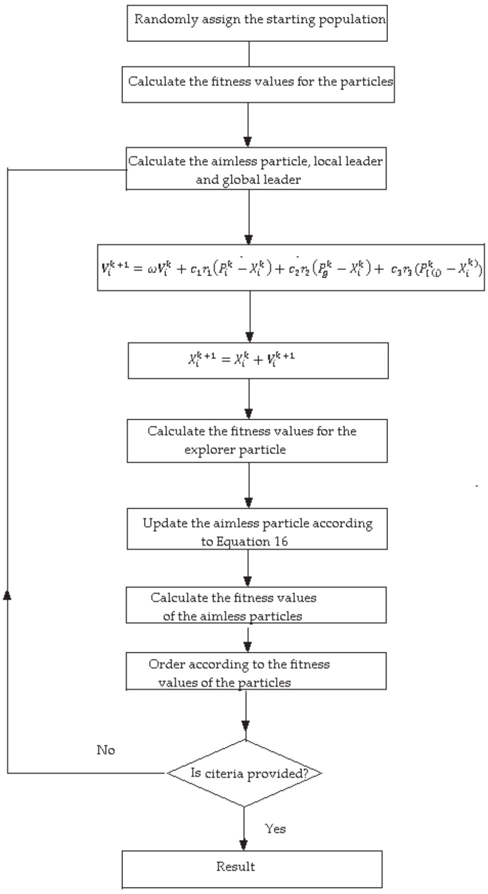
Randomly assign the starting population
Calculate the fitness values for the particles
Calculate the aimless particle, local leader and global leader
$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) + c_3 r_3 (P_{l(i)}^k - X_i^k)$
$X_i^{k+1} = X_i^k + V_i^{k+1}$
Calculate the fitness values for the explorer particle
Update the aimless particle according to Equation 16
Calculate the fitness values of the aimless particles
Order according to the fitness values of the particles
No
Is citeria provided?
Yes
Result

Figure 3 HPSSO flow diagram

3 PROPOSED METHOD

In this study, the micro grid consists of 3 energy sources: diesel generator, wind turbine and solar panel. The nominal power of the diesel generator is 4 MW and it supplies the residences when the renewable energy source is insufficient. Renewable energy farm consists of wind turbine and solar panels. The nominal power of the wind turbine is 4.5 MW, the nominal wind speed is 13.5 m/s, the maximum wind speed is 15 m/s. The wind turbine is disabled when the wind speed reaches the maximum speed. However, it continues to operate when the wind speed is nominal. The area covered by solar panels is 80000 m^2 and their efficiency is 10%. The micro grid feeds 1000 residences. Daytime consumption is low in the residence, while evening consumption is high.

The aim of this study is to investigate the reliability, emission and sizing of the energy sources of the analyzed micro grid. The aims of the simulation are explained as below:

- *RF* defines the energy amount by comparing RES with diesel generator. When the *RF* is 100%, it means that the demand is supplied completely by RES. *RF* provides information on the microgrid emission amount. The *RF* is calculated as given in Eq. (18) [34].

$$RF(\%) = \left(1 - \frac{\sum P_D}{\sum P_{PV} + P_{WT}}\right) \cdot 100 \tag{18}$$

- Reliability is an important factor related with load demand and it is the system that has sufficient electrical energy for meeting the required load demand. *LPSP* is defined by hybrid RES as the possibility to actualize the load demand. *LPSP* interval is between 0 and 1. A value of 1 for *LPSP* means that the hybrid RES cannot meet the required load demand. A value of 0 for *LPSP* indicates that hybrid RES can meet the load demand. *LPSP* is calculated using Eq. (19) [35].

$$LPS = \frac{\sum P_L - P_{PV} - P_{WT} + P_{SOC_{low}} + P_D}{\sum P_L} \tag{19}$$

- Sizing of the distribution energy resources does not only decrease power losses but also provides an increase in reliability. Minimum sizing of DER provides a more efficient and reliable electrical network [36]. Sizing formula is given in Eq. (20) [30].

$$\text{low}\left(P_L = \sum_{i=1}^{N} PG_i - P_D\right) \tag{20}$$

The constraints are given by the following equations:

$$\left(0.1 \times X_{lowlimit}\right) \le V\left(0.1 \times X_{upperlimit}\right) \tag{21}$$

$$X_{lowlimit} \le X \le X_{upperlimit} \tag{22}$$

$$0 \le c_1 \le 2 \tag{23}$$

$$0 \le c_2 \le 2 \tag{24}$$

$$0 \le c_3 \le 2 \tag{25}$$

$$0.4 \le \omega \le 0.9 \tag{26}$$

$$\lambda_{\max} < \lambda < \lambda_{\min} \tag{27}$$

The aim of this study was to increase the reliability of the microgrid, decrease its emission and size the energy resources at a minimum. Objective function was put forth in Eq. (28) in accordance with these objectives. Here, ω_1, ω_2 and ω_3 are the given coefficients and their sum should equal 1. The objective with greater importance should also have a higher coefficient. These values were selected as 0.6; 0.2 and 0.2 during simulations. Objective function formula is given in Eq. (28).

$$obj = \frac{1}{\left(0.6 \cdot LPSP\right) + \left(0.2 \cdot RF\right) + \left(0.2 \cdot PL\right)} \tag{28}$$

4 OPTIMIZATION RESULTS

4.1 Comparing the Iterations of PSO, SSO and HPSSO

In this application, PSO, SSO and HPSSO algorithms are compared in terms of different iterations. The effect of different number of iterations on the efficiency of the algorithms was investigated. The size of the population was selected as 100 in the algorithms. Microgrid consists of P_{PV} = 8000 kW, P_{WT} = 4500 kW and P_D = 15000 kW. This microgrid feeds of $P_{residence}$ = 10000 kW. In PSO algorithm for $n_{iteration}$ = 800 value, the best results of *LPSP* = 0.1392, *RF* = 0.9252 and *Psize* = 0.11 were obtained. As the *LPSP* value approaches zero, reliability increases. The *RF* value indicates the extent to which it benefits from RES. For $n_{iteration}$ = 400, PSO algorithm fed the network from the RES in 92,52%. As it is seen in Tab. 1 of PSO algorithm, as the number of iterations increased, reliability of the obtained microgrid increased and emission and sizing decreased. The increase in the number of iterations for the PSO algorithm led to better optimum results of the algorithm. In SSO algorithm, *LPSP* = 0, *RF* = 1 for all iterations. The reliability of the microgrid obtained for all iterations in the SSO algorithm is maximum, emission and sizing is minimum. In this algorithm, aimless particles have obtained high efficiency because they search outside of local search. The explorer has carried out aimless particle searches at points where the particles cannot search. In the HPSSO algorithm, the efficiency of the algorithm decreased as the number of iterations increased. This algorithm obtained the best value with *LPSP* = 0.1528, *RF* = 0.9173 and P_{isize} = 0,1 for $n_{iteration}$ = 100.

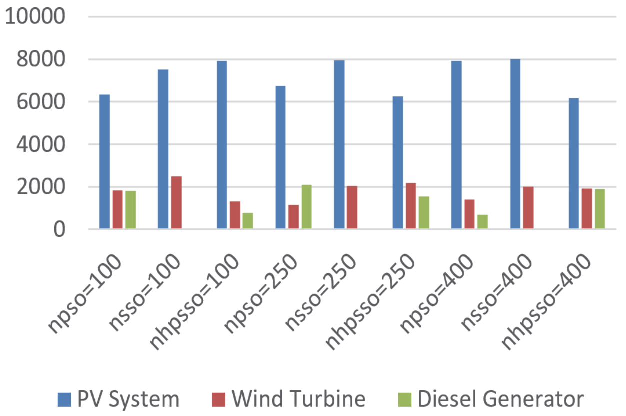

Figure 4 Energy resources of PSO, SSO, HPSSO algorithm fo $n_{iteration}$ = 100, 250, 400

The energy sources obtained with this algorithm are the cleanest. The PSO algorithm obtained P_{diesel} = 2095 KW at 250th iteration and has the highest emission value of the microgrid. The HPSSO algorithm achieved better results in the PSO algorithm for different iteration values. The HPSSO algorithm obtained the highest diesel generator power of the microgrid obtained for $n_{iteration}$ = 400 and produced the highest emission value.

Table 1 *LPSP*, *RF*, P_{size} of PSO, SSO and HPSSO for $n_{iteration}$ = 100, 250, 400

	$LPSP$	RF	P_{size}
$n_{PSO\ iteration} = 100$	0.3596	0.7808	0.21
$n_{SSO\ iteration} = 100$	0	1	0.6
$n_{HPSSO\ iteration} = 100$	0.1528	0.9173	0.1
$n_{PSO\ iteration} = 250$	0.419	0.7350	0.16
$n_{SSO\ iteration} = 250$	0	1	0.1
$n_{HPSSO\ iteration} = 250$	0.3106	0.8161	0.22
$n_{PSO\ iteration} = 400$	0.1392	0.9252	0.11
$n_{SSO\ iteration} = 400$	0	1	0
$n_{HPSSO\ iteration} = 400$	0.38	0.7654	0.22

Table 2 Power values of PSO, SSO and HPSSO for $n_{iteration}$ = 100, 250, 400

	P_{PV} / kW	P_{WT} / kW	P_{diesel} / kW
$n_{PSO\ iteration} = 100$	6356	1846	1798
$n_{SSO\ iteration} = 100$	7513	2487	0
$n_{HPSSO\ iteration} = 100$	7919	1317	764
$n_{PSO\ iteration} = 250$	6747	1158	2095
$n_{SSO\ iteration} = 250$	7951	2049	0
$n_{HPSSO\ iteration} = 250$	6268	2179	1553
$n_{PSO\ iteration} = 400$	7906	1398	696
$n_{SSO\ iteration} = 400$	8000	2000	0
$n_{HPSSO\ iteration} = 400$	6171	1929	1900

4.2 Population Comparison for PSO, SSO and HPSSO

In this application PSO, SSO and HPSSO algorithms were examined in terms of different population size. The algorithms were applied for the values of $n_{iteration} = 400$, $P_{PV} = 8000$ kW, $P_{WT} = 4500$ kW and $P_D = 15000$ kW. The population size influenced the efficiency of each algorithm differently. When the PSO algorithm population size is selected as small, exploratory particles in this algorithm failed to perform sufficient searches. As a result, İt was unable to achieve adequate improvement. In this algorithm, $LPSP = 0.1212$, $RF = 0.9355$ and $P_{size} = 0.15$ were the best values for $n_{population} = 60$. The microgrid obtained by the SSO algorithm for all population values has maximum reliability and minimum emission values. Maximum efficiency has been obtained by this algorithm compared to other algorithms. This algorithm obtained $LPSP = 0$ and $RF = 1$ for all population sizes. The HPSSO algorithm sticked around the local solutions as the population size grew and resulted in poor results. The best value of this algorithm obtained $LPSP = 0.3238$, $RF = 0.8068$ and $P_{size} = 0$ for $n_{population} = 20$. Fig. 5 shows the power values of the algorithms for different population sizes.

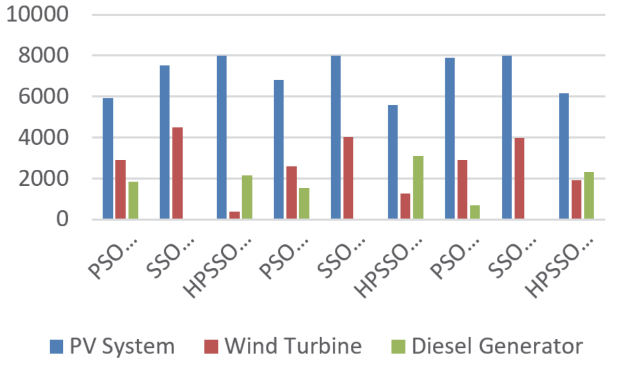

Figure 5 Energy resources of the PSO, SSO and HPSSO algorithms for $n_{population}$ = 20, 60, 100

As shown in Tab. 4, if the size of the population decreases in the PSO algorithm, the emission value of the obtained micro grid is increased and its reliability is decreased. When the population size of the SSO algorithm is increased, the micro-grid utilizes from the solar energy more than the wind energy. As seen in Tab. 3, HPSSO algorithm has the highest emission and lowest reliability value for the microgrid $P_{diesel} = 3136$ kW for $n_{population} = 60$. PSO algorithm gives better results than HPSSO algorithm.

Table 3 *LPSP*, *RF*, P_{size} of PSO, SSO and HPSSO $n_{population}$ = 20, 60, 100

	$LPSP$	RF	P_{size}
$n_{PSO\ iteration} = 20$	0.2308	0.8695	0.26
$n_{SSO\ iteration} = 20$	0	1	0.06
$n_{HPSSO\ iteration} = 20$	0.3238	0.8068	0
$n_{PSO\ iteration} = 60$	0.1212	0.9355	0.15
$n_{SSO\ iteration} = 60$	0	1	0
$n_{HPSSO\ iteration} = 60$	0.6272	0.5431	0.32
$n_{PSO\ iteration} = 100$	0.1392	0.9252	0.11
$n_{SSO\ iteration} = 100$	0	1	0
$n_{HPSSO\ iteration} = 100$	0.38	0.7654	0.22

Table 4 Power values of PSO, SSO and HPSSO for $n_{population}$ = 20, 60, 100

	P_{PV} / kW	P_{WT} / kW	P_{diesel} / kW
$n_{PSO\ iteration} = 20$	5933	2913	1154
$n_{SSO\ iteration} = 20$	7505	2495	0
$n_{HPSSO\ iteration} = 20$	8000	381	1619
$n_{PSO\ iteration} = 60$	6794	2600	606
$n_{SSO\ iteration} = 60$	8000	2000	0
$n_{HPSSO\ iteration} = 60$	5585	1279	3136
$n_{PSO\ iteration} = 100$	7906	1398	696
$n_{SSO\ iteration} = 100$	8000	2000	0
$n_{HPSSO\ iteration} = 100$	6171	1929	1900

4.3 Variable Load Analysis for the SSO Algorithm with the Best Results

The reliability, emission and sizing of the microgrid at 25%, 50%, 75% and 100% load were examined in this application. For this purpose, the SSO algorithm was used which yielded the best results in previous applications. Fig. 6 shows the power values for different load values. The data for the application are provided below.

$n_{iteration} = 400$
$P_{residence\ load} = 10000$ kW
$P_{solar\ \mathrm{power\ plant}} = 5000$ kW
$P_{wind\ \mathrm{turbine}} = 2000$ kW
$P_{diesel\ \mathrm{generator}} = 4000$ kW

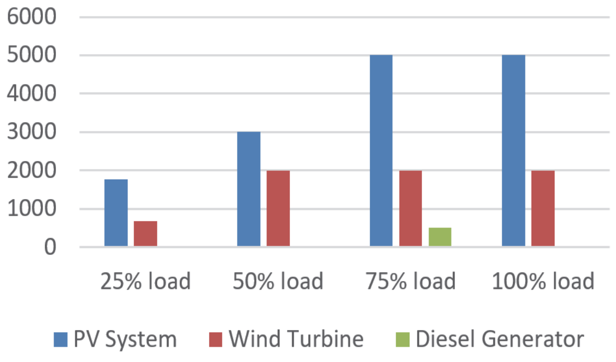

Figure 6 Load analysis for the SSO algorithm

Table 5 *LPSP*, *RF*, P_{size} of the SSO algorithm for variable loads

	$LPSP$	RF	P_{size}
0 - 25% load	0	0	0,038
26 - 50% load	0	0	0,001
51 - 75% load	0.068	0.015	0,5
76 - 100% load	0.39	0.086	0,21

Table 6 *LPSP*, *RF*, P_{size} of the SSO algorithm for variable loads

	P_{PV}	P_{WT}	P_{diesel}
0 - 25%load	1771	690	0
26 - 50% load	3000	2000	0
51 - 75% load	5000	2000	500
76 - 100% load	5000	2000	955

Tab. 6 shows that the solar power generator and the wind turbine supply sufficient energy to the residence at loads of 25% and 50% and the diesel generator is not used. For these loads, $LPSP = 0$ meaning that reliability is at the maximum level. Moreover, emission is zero and sizing is at a minimum. As can be seen in Tab. 5, reliable energy was obtained with the SSO algorithm at loads of 75%, 100%. In addition, the energy sources obtained with this algorithm produced minimum results for emission and sizing.

5 CONCLUSIONS

In the scientific studies conducted with microgrid, no research has been found about SSO and HPSSO. Therefore, the effect of these algorithms on the microgrid is investigated. Possible scenarios were developed with PSO, SSO, and HPSSO algorithm under different iteration values, different population sizes, and varying load conditions in order to improve the analysed microgrid. Simulation results showed that SSO algorithm achieved better results than other optimization methods. This algorithm obtained $LPSP = 0$ and $RF = 1$ values for all population and iteration values. This algorithm has achieved better optimum values and high convergence rate. When the SSO algorithm is analysed with different iteration and population values, the reliability of the obtained microgrid is maximum and emission is minimum. In addition, the performance of SSO algorithm over 50% load and over 50% load was examined. The energy sources obtained with this algorithm have achieved minimum emissions up to 50% load, maximum reliability and good results above 50% load. PSO algorithm had better results than HPSSO algorithm. In studies conducted in the literature, HPSSO algorithm has obtained better results than PSO and SSO. However, SSO algorithm performed better in this study. The performance of these algorithms may change if the micro grid is examined from a different perspective. SSO algorithm cannot be claimed to be the best optimization technique. But it has achieved good values.

The SSO algorithm can be applied to many engineering problems. The emission amount, sizing and reliability of the sources can be calculated with the proposed algorithm before the microgrid installation is realized. As a result, the cost of the microgrid is reduced and its performance increases.

6 REFERENCES

[1] Lee, Y., D., Jiang, J. L., Su, H. J., Ho, Y. H., & Chang, Y. R. (2016). Ancillary voltage control for a distribution feeder by using energy storage system in microgrid ın power electronics for distributed generation systems (pedg). *2016 IEEE 7th International Symposium On.* https://doi.org/10.1109/PEDG.2016.7527068

[2] Zacharia, L., Kyriakou, A., Hadjidemetriou, L., Kyriakides, E., Azzopardi, C. P. B., Martensen, N., & Borg, N. (2018). Islanding and resynchronization procedure of a university campus microgrid. *In 2018 International Conference OnSmart Energy Systems and Technologies,* https://doi.org/10.1109/SEST.2018.8495828

[3] Zheng, Y., Song, Y., Hill, D. J., & Zhang, Y. (2018). Multiagent system based microgrid energy management via asynchronous consensus admm. *IEEE Transactions On Energy Conversion*, *33*(2), 886-888. https://doi.org/10.1109/TEC.2018.2799482

[4] Ou, T. C. & Hong, C. M. (2014). Dynamic operation and control of microgrid hybrid power systems. *Energy*, *66*, 314-323. https://doi.org/10.1016/j.energy.2014.01.042

[5] Eriksson, E. L. V. & Gray, E. M. (2017). Optimization and ıntegration of hybrid renewable energy hydrogen fuel cell energy systems-a critical review. *Applied Energy*, *202*, 348-364. https://doi.org/10.1016/j.apenergy.2017.03.132

[6] Askarzadeh, A. (2016). A Novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers & Structures*, *169*, 1-12. https://doi.org/10.1016/j.compstruc.2016.03.001

[7] Gafar, M. G., Elhoseny, M., & Gunasekaran, M. (2020). Modeling neutrosophic variables based on particle swarm optimization and information theory measures for forest fires. *The Journal of Supercomputing*, *76*(4), 2339-2356. https://doi.org/10.1007/s11227-018-2512-5

[8] Xia, X., Gui, L., He, G., Wei, B., Zhang, Y., Yu, F., & Zhan, Z. H. (2020). An expanded particle swarm optimization based on multi-exemplar and forgetting ability. *Information Sciences*, *508*, 105-120. https://doi.org/10.1016/j.ins.2019.08.065

[9] Sarangi, S. K., Panda, R., & Abraham, A. (2020). Design of optimal low-pass filter by a new Levy swallow swarm algorithm. *Soft Computing*, *24*, 8113-18128. https://doi.org/10.1007/s00500-020-05065-6

[10] Bouzidi, S., Bouzidi, M., & Riffi, M. E. (2019). Solving the Quadratic Assignment Problem using the Swallow Swarm Optimization Problem. *International Journal of Engineering and Advanced Technology (IJEAT)*, 3116-3120. https://doi.org/10.35940/ijeat.F9132.088619

[11] Kaveh, A., Bakhshpoori, T., & Afshari, E. (2014). An Efficient Hybrid Particle Swarm and Swallow Swarm Optimization Algorithm. *Computers & Structures*, *143*, 40-59. https://doi.org/10.1016/j.compstruc.2014.07.012

[12] Azaza, M. & Wallin, F. (2017). Multi objective particle swarm optimization of hybrid micro-grid system: A case study in Sweden. *Energy*, *123*, 108-118. https://doi.org/10.1016/j.energy.2017.01.149

[13] Bukar, A. L., Tan, C. W., & Lau, K. Y. (2019). Optimal sizing of an autonomous photovoltaic/wind/battery/diesel generator microgrid using grasshopper optimization algorithm. *Solar Energy,* 188, 685-696. https://doi.org/10.1016/j.solener.2019.06.050

[14] Kerdphol, T., Fuji, K., Mitani, Y., Watanabe, M., & Qudaih, Y. (2016). Optimization of a battery energy storage system using particle swarm optimization for stand-alone microgrids. *International Journal of Electrical Power & Energy Systems*, *81*, 32-39. https://doi.org/10.1016/j.ijepes.2016.02.006

[15] Trivedi, I. N., Bhoye, M., Bhesdadiya, R. H., Jangir, P., Jangir, N., & Kumar, A. (2016). An emission constraint environment dispatch problem solution with microgrid using Whale Optimization Algorithm. *In 2016 National Power Systems Conference* (*NPSC*), 1-6. https://doi.org/10.1109/NPSC.2016.7858899

[16] Wang, Y., Li, F., Yu, H., Wang, Y., Qi, C., Yang, J., & Song, F. (2020). Optimal operation of microgrid with multi-energy complementary based on moth flame optimization algorithm. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, *42*(7), 785-806. https://doi.org/10.1080/15567036.2019.1587067

[17] Javid, Z., Li, K. J., Ul Hassan, R., & Chen, J. (2020). Hybrid-Microgrid Planning, Sizing and Optimization for an Industrial Demand in Pakistan. *Tehničkivjesnik*, *27*(3), 781-792. https://doi.org/10.17559/TV-20181219042529

[18] Bouzidi, S. & Riffi, M. E. (2017). Discrete swallow swarm optimization algorithm fortravelling salesman problem. *In Proceedings of the 2017 International Conference on Smart Digital*, 80-84, Rabat Morocco. https://doi.org/10.1145/3128128.3128141

[19] Revathi, K. & Krishnamoorthy, N. (2015). The Performance Analysis of Swallow Swarm Optimization Algorithm. In 2015 *2nd International Conference on Electronics and Communication Systems* (*ICECS*), *26-27*, 558-562. https://doi.org/10.1109/ECS.2015.7124970

[20] Neshat, M., Sepidnam, G., & Sargolzaei, M. (2013). Swallow Swarm Optimization Algorithm: A New Method to Optimization. *Neural Computing and Applications*, *23*, 2, 429-454. https://doi.org/ 10.1007/s00521-012-0939-9

[21] Sam, C. C. & Ali, A. A. (2015). Identification of Crack in A Cantilever Beam Using Improved PSO Algorithm. *Interantional Journal for Innovative Research in Science & Technology*. 454-461. http://www.ijirst.org/articles/IJIRSTV1I11187.pdf.

[22] Kaveh, A., Bakhshpoori, T., & Afshari, E. (2015). Hybrid PSO And SSO Algorithm for Truss Layout and Size Optimization Considering Dynamic Constraints. *Structural Engineering and Mechanics*., *54*(3), 453-474. https://doi.org/10.12989/sem.2015.54.3.453

[23] Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers & Structures*, *169*, 1-12. https://doi.org/10.1016/j.compstruc.2016.03.001

[24] Gandomi, A. H., Yang, X. S., Talatahari, S., & Alavi, A. (2013). Metaheuristic algorithms ınmodeling and optimization. *In Metaheuristic Applications In Structures And Infrastructures*, 1-24.

[25] Yi, H., Duan, Q., & Liao, T. W. (2013). Three ımproved hybrid metaheuristic algorithms for engineering design optimization. *Applied Soft Computing*, *13*(5), 2433-2444. https://doi.org/10.1016/j.asoc.2012.12.004

[26] Cheng, M. Y., &Prayogo, D. (2014), Symbiotic organisms search: a new metaheuristic optimization algorithm. *Computers & Structures*, 139, 98-112, https://doi.org/10.1016/j.compstruc.2014.03.007

[27] Gong, Y. J., Li, J. J., Zhou, Y., Li, Y., Chung, H. S. H., Shi, Y. H., & Zhang, J. (2016), Genetic learning particle swarm optimization. *IEEE Transactions On Cybernetics*, *46*(10), 2277-2290. https://doi.org/10.1109/TCYB.2015.2475174

[28] Lee, C. S., Ayala, H. V. H., & dos Santos Coelho, L. (2015), Capacitor placement of distribution systems using particle swarm optimization approaches. *International Journal of Electrical Power & Energy Systems*, *64*, 839-851, https://doi.org/10.1016/j.ijepes.2014.07.069

[29] Fu, H., Li, Z., Liu, Z., & Wang, Z. (2018), Research on big data digging of hot topics about recycled water use on micro-blog based on particle swarm optimization. *Sustainability*, *10*(7), 2488. https://doi.org/10.3390/ su10072488

[30] Carneiro, T. C., Melo, S. P., Carvalho, P. C., & Braga, A. P. D. S. (2016). Particle Swarm Optimization method for estimation of Weibull parameters: A case study for the Brazilian northeast region. *Renewable Energy*, *86*, 751-759. https://doi.org/10.1016/j.renene.2015.08.060

[31] Taherkhani, M. & Safabakhsh, R. (2016). A novel stability-based adaptive inertia weight for particle swarm optimization. *Applied Soft Computing*, *38*, 281-295, https://doi.org/10.1016/j.asoc.2015.10.004

[32] Neshat, M. & Sepidname, G. (2015). A New Hybrid optimization method ınspired from swarm ıntelligence: fuzzy adaptive swallow swarm optimization algorithm (fasso). *Egyptian Informatics Journal*, *16*(3), 339-350. https://doi.org/10.1016/j.eij.2015.07.003

[33] Varol, E. & Alataş, B. (2017). Sürüzekâsındayenibiryaklaşım: kuşsürüsüalgoritması., DÜMF. *Engineeering of Journal*, *8*(1), 133-146. http://www.dicle.edu.tr/muhendislikdergisi/cilt8sayi1/MD-16-059.pdf

[34] Upadhyay, S. & Sharma, M. P. (2015), Development of hybrid energy system with cycle charging strategy using particle swarm optimization for a remote area in India. *Renewable Energy*, *77*, 586-598. https://doi.org/10.1016/j.renene.2014.12.051

[35] Sawle, Y., Gupta, S. C., & Bohre, A. K. (2018). Socio-Techno-Economic design of hybrid renewable energy system using optimization techniques. *Renewable Energy*, *119*, 459-472. https://doi.org/10.1016/j.renene.2017.11.058

[36] Abdmouleh, Z., Gastli, A., Ben-Brahim, L., Haouari, M., & Al-Emadi, N. A. (2017). Review of optimization techniques applied for the integration of distributed generation from renewable energy sources. *Renewable Energy*, *113*, 266-280. https://doi.org/10.1016/j.renene.2017.05.087

Abbreviation	
SSO	Swallow swarm optimization
HPSSO	Hybrid particle swallow swarm optimization
PSO	Particle swarm optimization
RF	Renewable factor
LPSP	Loss of power supply probability
RES	Renewable energy sources
DER	Distributed energy resources
GOA	Grasshopper optimization algorithm
WOA	Whale optimization algorithm
Variables	
$f(X)$	Objective function
X	n dimensional design vector
$g_j(x)$	İnequality constraints
$l_j(x)$	Equality constraints
n	Number of variable
m	Number of inequality constraints
p	Number of equality constraints
ω	İnertia weight
r_1, r_2	Random values between 0 and 1
c_1, c_2	Cognitive and social components
p_{best}	Best position of the particle
p_{best}	Best position of the swarm
v_i	Particle velocity
x_i	Particle position
$V_{HL_{i+1}}$	Head leader velocity
α_{LL}	Acceleration coefficient of the local leader
β_{LL}	Acceleration coefficient for the local leader
LL_i	Local leader
α_{HL}	Acceleration coefficient of the head leader
rand	0 - 1 random number
e_{best}	Best value for the explorer particle
e_i	Explorer particle
β_{HL}	Acceleration coefficient for the head leader
HL_i	Head leader
V_{LL_i}	Velocity of the local leader

V_{i+1}	Velocity of the explorer particle
X_i^{k+1}	Updated position of the particle
X_i^k	Previous position of the particle
V_i^{k+1}	Updated velocity o of the particle
r_1, r_2, r_3	Represent a random number in the interval (0, 1)
c_1, c_2, c_3	Learning factor
$P_{l_i}^k$	Local leader of the sub-colony that the ith particle contains
$rand\,(-1,1)$	Random number between 1 and −1
$\min_s$, $\max_s$	Lower and upper limits of the design variables
$\lambda_{\max}$, $\lambda_{\min}$	First and last iteration values of the algorithm
P_{PV}	Power of the solar power plant
P_{WT}	Power of wind power
P_D	The power of the diesel power plant
$P_{residence}$	Residence power
P_L	Load power
$P_{soc_{low}}$	Battery power
$n_{iteration}$	Iteration number
$n_{population}$	Population number

Contact information:

Tuba TANYILDIZI AĞIR
(Corresponding author)
Department of Electrical and Electronic Engineering, Batman University,
Batman, Turkey
E-mail: tuba.tanyildiziagir@batman.edu.tr

Zafer AYDOĞMUŞ
Department of Electrical and Electronic Engineering, Firat University,
Elazig, Turkey
E-mail: zaydogmus@firat.edu.tr

Bilal ALATAŞ
Department of Software Engineering, Firat University, Elazig, Turkey
E-mail: balatas@firat.edu