

Control of Power Prosumer Based on Swarm Intelligence Algorithms

Pavel Matrenin^{1*}, Vadim Manusov¹, and Dmitry Antonenkov¹

¹Department of Industrial Power Supply Systems, Novosibirsk State Technical University, Novosibirsk, Russia

Abstract. The development of renewable energy and Smart Grid leads to the emergence of prosumers or power generating consumers, which are involved in the processes of bidirectional exchange of electricity and information. The work is devoted to the problem of optimal control of a power generating consumer in Smart Grid. The distinctive research features are the solution of the optimal control problem in conditions of difficult prediction of wind power plant generation, the usage of Swarm Intelligence algorithms to build a system of control rules, and the study of the obtained models on data from two different generating consumers: one on about Russky Island, the second on Popov Island (Far East). We selected a list of priority rules as a decision-making model and applied Particle Swarm Optimization, Bees Algorithm, and Firefly Optimization to build and optimize this model. The computer modeling with the usage of two mounts dataset showed that the proposed approach could significantly increase the revenue of the generating consumers considered.

1 Introduction

The development of renewable energy and Smart Grid leads to the emergence of prosumers or power generating consumers (GC), which are involved in the processes of bidirectional exchange of electricity and information [1, 2]. GC needs to control not only electrical load but also the flow of generated power. It significantly increases the complexity of its control tasks [3, 4].

The problem of the optimal GC control has a number of issues that lead to high complexity:

- GC operates under conditions of stochastic change in the generation of electricity by renewable sources and, to a lesser extent, of its consumption;
- the control problem has a high dimensionality of the solution search space;
- the objective function is not an analytical expression, is need to be calculated algorithmically.

Much modern research has been devoted to optimal control in Smart Grid networks with distributed generation and renewable energy sources. However, the optimal control is carried out at the level of a supersystem in them, and not individual GC. The frameworks to real-time coordinate load scheduling, sharing, trading were considered at studies [5, 6]. A.C. Luna et al. [7] proposed an energy management system for coordinating the operation of distributed household prosumers with renewable energy sources. H. Mortaji et al. [8] proposed smart-direct load control and load shedding based on autoregressive integrated moving average time-series prediction model and Internet of Things concept.

A number of articles propose a stochastic game approach for the problem of energy trading between smart grid prosumer. L. Ma et al. [6] used the energy management model on cooperative game theory. S.R. Etesami et al. [9] formulated the interaction among prosumers as a stochastic game, in which each prosumer seeks to maximize its payoff, in terms of revenues and proposed an optimal strategy for utility companies. The Stackelberg game approach for Smart Grid Energy Management (Energy sharing management) was considered at [10, 11].

Such management allows taking into account data on all participants in the distributed electric power system, but there is a risk associated with the high level of centralization of the control system. Thus, modern studies primarily consider the principles of constructing the entire Smart Grid power system and the interaction rules for multiple GCs. Our research focuses on optimizing the control rules for a single GC with a difficult prediction of generation using Swarm Intelligence (SI) algorithms.

The SI algorithms are known to effectively solve large-scale nonlinear optimization problems, including problems of power systems. The most commonly used SI algorithm is Particle Swarm Optimization (PSO); paper [12] provides a comprehensive survey on the usage PSO for power system applications. Other SI algorithms are also applied to different optimization problems in power system design and control [13-15]. In this research, three SI algorithms: PSO, Bees algorithm (BA), and Firefly optimization (FFO).

* Corresponding author: pavel.matrenin@gmail.com

2 The Problem Statement

2.1 GC Power System

In this research, we considered two large GC: the power system of Russky Island and the power system of Popov Island. Both islands are located in Peter the Great Gulf in the East Sea (Fig. 1). High wind speed makes it possible to create wind power plants up to 16 MW on Russky Island and up to 20 MW on Popov Island [16].

Russky Island belongs to the territorial composition of Vladivostok. It is located about two kilometers from the coast in Peter the Great Gulf, which is part of the Sea of Japan (the smallest distance between the continental part and the island is 800 meters). Russky Island is separated from the Muravyov-Amursky Peninsula by the East Bosphorus. From the west, the island is washed by the waters of the Amur Gulf, and from the east and south side by the waters of the Ussuri Gulf. In the southwest, the island is separated from the other Popov island by the Stark Strait.

The island is 97.6 km², its length is about 18 km, and its width is about 13 km. The population of the island is approximately 25,000 inhabitants.

Popova Island (named after Admiral A.A. Popova) is located in Peter the Great Gulf of the Sea of Japan, 20 km from Vladivostok and 0.5 km southwest of Russky Island. About 3,000 people live on the island, mainly in the two villages of Stark and Popova.

Fig. 2 and Fig 3. show curves of own consumption of Russky GC and Popova GC and possible wind power generation of GCs according to the estimates of [16]. Fig 4. shows the results of adding these curves. The demonstrated fragment of data corresponds to twenty days from 01.06.2017.

From the charts above it is possible to notice the following:

- the forms of the electricity generation curves are very close since both islands are located very close, and their wind speeds are also close;
- the load curves concerning to generation are radically different; the Russky GC always has a deficit, the Popova GC still has a surplus;
- in case of consideration of two consumers together, we have the third type of load / generation pattern – in general, the GC system has a deficit, but sometimes there is a surplus.

Thus, different management strategies may be required depending on the characteristics of GC or GC hub.

2.2 Optimal Control

The task of optimal control is to create a control system (subject) that implements a sequence of actions on a controlled object in the environment to achieve the best possible quality specified by one or more criteria (Fig. 5); the controlled object is a specific part of the world around which the control subject can purposefully influence [17]. Control always occurs during a certain period of time, while the controlled object passes from one state to another.

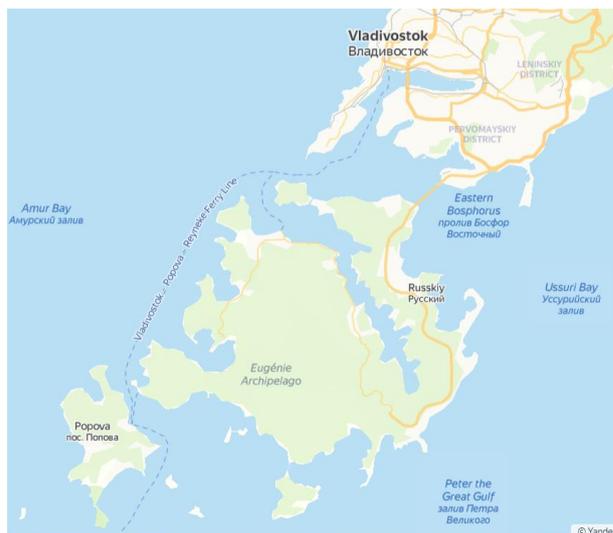


Fig. 1. Russky and Popova Islands.

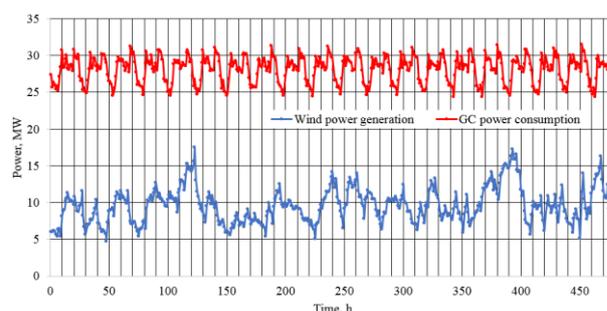


Fig. 2. Consumption and generation of Russky GC.

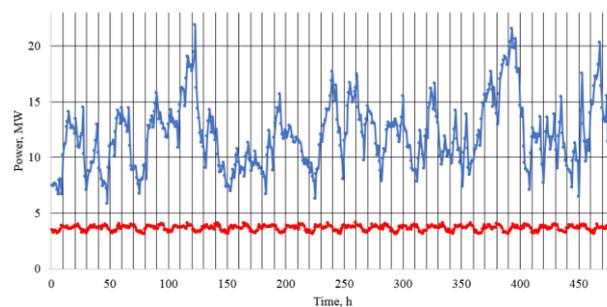


Fig. 3. Consumption and generation of Popova GC.

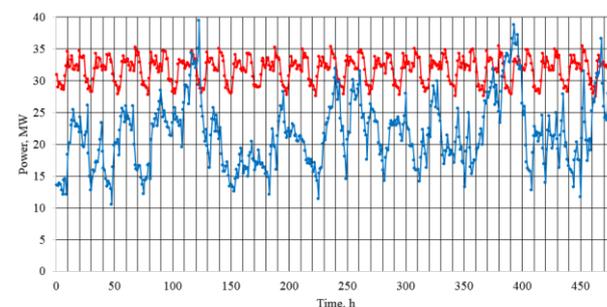


Fig. 4. Aggregated consumption and generation of both GC.

The state of the controlled object is characterized by a set of parameters that can change over time:

$$S(t) = \{s_1(t), s_2(t), \dots, s_n(t)\}. \quad (1)$$

Thus, there is a vector of functions. Each function shows the parameter changing over time. These functions in the

explicit form are unknown. In addition, there is a control system that provides control.

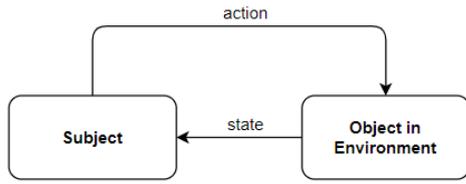


Fig. 5. Interaction between a subject and an object.

The control can also be defined as a vector of functions:

$$A(t) = \{a_1(t), a_2(t), \dots, a_m(t)\}. \quad (2)$$

The notations S from “state” and A from “action” are used.

The optimal control problem, in general, can be written as follows:

$$A^{opt}(t) = \arg \min_{A(t) \in A^{pos}} \int_{t_0}^{t_T} f(t, S(t), A(t)) dt \quad (3)$$

- $A^{opt}(t)$ is the required optimal control; it defines values of the control parameters at each time moment (in the considered task, when and how much GC must sell or buy, charge or discharge);
- A^{pos} is the area of permissible values of control parameters;
- $f(t, S(t), A(t))$ is a continuous-time cost function, in the considered task, it gets GC’s total electricity costs: purchases from the own generation + purchases from the other GC and an external power system – sale to other GC and the external power system.
- t_0 and t_T are the period of time considered.

2.3 GC Optimal Control task

For GC, the state parameters can be defined as follows:

- own consumption, MWh (s_1);
- wind power plants generation, MWh (s_2);
- charge level of power storage, MWh (s_3).

Control parameters can be defined as follows:

- the amount of electricity that is currently exchanged by the GC with an external power system (purchase or sale), MWh (a_1);
- the amount of electricity that is currently being transferred by the GC with the neighboring GC (purchase or sale), MWh (a_2);
- the amount of electricity that the GC is currently charging or discharging from the power storage, MWh (a_3).

The control does not affect the state parameters associated with the GC consumption and generation, but it directly affects the charge of the power storage. In the considered task, the time step is set equal to one hour. So, each day contains 24 values of the state parameters and 24 values of the control parameters. A daily sample is shown in Fig. 6.

Due to the high complexity of power systems in an explicit analytical form, the function $f(t, S(t), A(t))$

cannot usually be obtained, especially integral of this function. But it is possible to calculate the function algorithmically. In the case of GC control, this function is piecewise continuous, since the time step is 1 hour. The task (3) can be written without an integral, in the form of a sum, and the function $f(t, S(t), A(t))$ is nothing more than the difference between the revenues from the sale of electricity of a GC and the costs of its purchase, generation, and power storage in all hours into the time period. However, even in this case, the analytical expression for $f(t, S(t), A(t))$ is difficult to write, since the price of electricity is a piecewise constant function, the exchange of electricity with a neighboring GC supply depends on its state and controlling them. Thus, the calculation of the value of $f(t, S(t), A(t))$ should be performed algorithmically:

$$A^{opt}(t) = \arg \min_{A(t) \in A^{pos}} \sum_{t=t_0}^T revenue(t, S(t), A(t)) \quad (4)$$

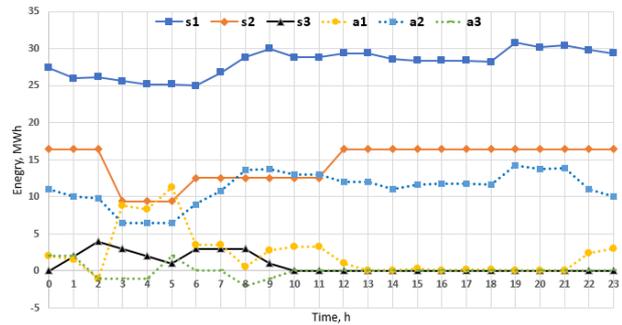


Fig. 6. Sample of daily state-action curves of Russian GC.

3 The Research Method

3.1 Heuristic-Rule-based control

All possible control actions could be described by dividing them into four groups. The following designation is used:

- *power_wind* – GC wind power plant generation at the considered hour;
- *power_gc* – GC consumption at the considered hour;
- *dif* – the difference between the GC generation and consumption at the considered hour;
- *storage* – the amount of energy that needs to be charged (> 0) or discharged (<0) at the considered hour;
- *now_storage* – the energy stored in the power storage at the considered hour;
- *max_storage* – the maximum amount of energy that can be stored in the power storage (constant, GC parameter);
- *max_storage_h* – the maximum amount of energy that can be added to the power storage in one hour (constant, GC parameter);
- *sale_storage* – coefficient that regulates the balance of purchase and charging (parameter should be tuned in the optimization process)
- *sale_unload* – coefficient that regulates the balance of sales and use of discharging (parameter should be tuned in the optimization process);

- *sale_buy* – the amount of energy that is sold (> 0) or purchased (<0) at the considered hour.

Thus, we have the four cases of possible control actions:

1. Charge_Sell (it's possible if generation > consumption):
 - 1.1. $dif = power_wind - power_gc$;
 - 1.2. $storage = \min(max_storage - now_storage, max_storage_h, dif)$;
 - 1.3. $storage = storage * sale_storage$;
 - 1.4. $now_storage = now_storage + storage$;
 - 1.5. $sale_buy = dif - storage$.
2. Charge_Buy:
 - 2.1. $dif = power_wind - power_gc$;
 - 2.2. $storage = \min(max_storage - now_storage, max_storage_hour)$;
 - 2.3. $storage = storage * buy_storage$;
 - 2.4. $now_storage = now_storage + storage$;
 - 2.5. $sale_buy = dif - storage$.
3. Discharge_Sell:
 - 3.1. $dif = power_wind - power_gc$;
 - 3.2. $storage = now_storage$;
 - 3.3. $storage = storage * sale_unload$;
 - 3.4. $now_storage = now_storage - storage$;
 - 3.5. $sale_buy = dif + storage$.
4. Discharge_Buy (it's possible if generation < consumption):
 - 4.1. $dif = power_wind - power_gc$;
 - 4.2. $storage = \min(-dif, now_storage)$;
 - 4.3. $storage = storage * buy_unload$;
 - 4.4. $now_storage = now_storage - storage$;
 - 4.5. $sale_buy = storage - dif$.

The choice of actions should depend on the state of the GC, but it is enough to get answers to two questions. The first is connected with determining whether the GC is in a state of excess or deficiency of energy? The second is also related to the fact that the price of electricity changes throughout the day. Although various billing schemes are possible, a two-zone tariff is considered in this research, the daily tax is from 7 a.m. to 11 p.m., and at other hours it is a night tax, cheaper one. Thus, it's needed to get answers to the questions:

- 1) Excluding accumulation, does the generation of the GC wind power plant more than the GC consumption ($dif > 0$)?
- 2) Is there a special time period now?

The GC control takes into account the possibility of using two intervals as special periods (from $time_1$ to $time_2$ and from $time_3$ to $time_4$), the values of the boundaries of the time intervals are parameters adjusted during the optimization process.

As a result, we have four possible cases at each hour:

- ($dif < 0$) AND NOT (special_time_period);
- ($dif > 0$) AND NOT (special_time_period);
- ($dif < 0$) AND (special_time_period);
- ($dif > 0$) AND (special_time_period).

When creating a GC control based on rules, we get 12 rules of the form IF <condition>, THEN <action>

The number of rules is 12 since the second and third actions can be performed under any of the four conditions, and the first and fourth under two conditions ($2 * 4 + 2 * 2 = 12$). In addition, the GC control model

has four balance factors: *buy_unload*, *sale_unload*, *buy_storage*, *sale_storage*, and four moments as the boundaries: $time_1, time_2, time_3, time_4$.

To control using these rules, we need to determine the procedure for their verification and compliance, that is, rule priorities. Decision making begins with checking of the highest priority rule. If its condition is satisfied, then the corresponding action of this rule is implemented. Otherwise, the next priority rule is checked, and so on until the end of the rule list. The conditions are designed in such a way that when you go through the list of rules, you will surely find one whose condition will be satisfied. As a result, to build a controller, it is necessary to determine the order of the rules by setting priorities (pr_i) and the tuned parameters specified above:

$$Solution = [pr_1, \dots, pr_{12}, buy_unload, sale_unload, buy_storage, sale_storage, time_1, \dots, time_4]$$

3.2 Swarm Intelligence

It is not always possible to determine the Swarm Intelligence algorithm that is most suitable for a solved task [15]. Therefore, the use of only one algorithm can give a solution whose effectiveness is not satisfactory for the optimization criterion. In this case, the researcher cannot determine the effectiveness without using other algorithms for comparison. Therefore, three Swarm Intelligence algorithms were applied: the Particle Swarm Optimization (PSO) algorithm, the Firefly Optimization (FFO) algorithm, and the Bees algorithm (BA) (not Artificial Bee Colony Optimization). Descriptions of the algorithms precisely in the form in which they are applied in this research are given in.

3.2.1 Particle Swarm Optimization

The Particle Swarm Optimization algorithm was first proposed by J. Kennedy and R. Eberhart in 1995 [18]. Then it was improved by Kennedy, Eberhart, and Shi [19]. PSO is based on a bird flocks' behavior. A flock acts coordinated according to a number of simple rules. Every bird (called 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 a number of 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 algorithms description [15], the PSO algorithm may be represented by a tuple $\{S, M, A, P, I, O\}$.

1. A set of particles (particles) $S = \{s_1, s_2, \dots, s_{|S|}\}$, $|S|$ is number of particles. At j -th iteration i -th particle is characterized by the state $s_{ij} = \{X_{ij}, V_{ij}, X_{ij}^{best}\}$, where $X_{ij} = \{x_{ij}^1, x_{ij}^2, \dots, x_{ij}^l\}$ is the variable parameter vector (particle position), $V_{ij} = \{v_{ij}^1, v_{ij}^2, \dots, v_{ij}^l\}$ is the velocity vector,

$X_{ij}^{best} = \{b_{ij}^1, b_{ij}^2, \dots, b_{ij}^l\}$ 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.

2. Means of indirect exchange is vector $M = X_{ij}^{best}$ is the best value of the variable parameters vector derived among all particles from the 1st to the j -th iterations of the algorithm.

3. Algorithm A describes the steps of the PSO algorithm.

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

$$\begin{aligned} X_{ij} &\leftarrow \text{random}(0,1), i = 1, \dots, |S|, \\ V_{ij} &\leftarrow \text{random}(0, v_{max}), i = 1, \dots, |S|, \\ X_{ij}^{best} &\leftarrow X_{ij}, i = 1, \dots, |S|, \end{aligned}$$

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

3.2. 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 $\{I, O\}$.

$$\begin{aligned} X_{ij}^{best} &\leftarrow X_{ij} | f(X_{ij}^{best}) < f(X_{ij}), i = 1, \dots, |S|, \\ M &\leftarrow X_{ij} | f(M) < f(X_{ij}), i = 1, \dots, |S|, \end{aligned}$$

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

$$V_{ij+1} \leftarrow V_{ij}\omega + \alpha_1(X_{ij}^{best} - X_{ij})\text{random}(0,1) + \alpha_2(M - X_{ij})\text{random}(0,1) \quad (i = 1, \dots, |S|).$$

$$\begin{aligned} V_{ij+1} &\leftarrow \beta | V_{ij+1} > v_{max}, i = 1, \dots, |S|, \\ V_{ij+1} &\leftarrow -\beta | V_{ij+1} < -v_{max}, i = 1, \dots, |S|, \end{aligned}$$

where $\alpha_1, \alpha_2, \omega, v_{max}$ are algorithm parameters.

$$\begin{aligned} X_{ij+1} &\leftarrow X_{ij} + V_{ij+1}, i = 1, \dots, |S|, \\ X_{ij+1} &\leftarrow 1 | X_{ij+1} > 1, i = 1, \dots, |S|, \\ X_{ij+1} &\leftarrow 0 | X_{ij+1} < 0, i = 1, \dots, |S| \end{aligned}$$

3.4. 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.

4. 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 v_{max} defines limitations for the maximum velocity.

5. Identifiers I and O are input and output of the PSO algorithm for interaction with the problem considered.

3.2.2 Bees Algorithm

The Bees Algorithm Bee Colony Optimization algorithm was researched and developed by a number of authors in 2005 [20]. It is based on the simulation of the behaviour 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. 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, the BA algorithm may be represented by a tuple $\{S, M, A, P, I, O\}$.

1. A set of particles (bees) $S = \{s_1, s_2, \dots, s_{|S|}\}$. At the j -th iteration the i -th particle is characterized by the state $s_{ij} = \{X_{ij}\}$, where $X_{ij} = \{x_{ij}^1, x_{ij}^2, \dots, x_{ij}^l\}$ is the variable parameters vector (the particle position), l is the dimensionality of the solution search space.

2. Means of indirect exchange M is a list of the best and perspective positions found in the j -th iteration, $M = \{N_{ij}^b, N_{kj}^g\}, i = 1, \dots, n^b, k = 1, \dots, n^g$.

3. Algorithm A describes the steps of the BA algorithm.

3.1. Generation of initial population ($j=1$) is fulfilled only for a subset of particles termed scouts:

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

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

3.2. 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 $\{I, O\}$.

$$X_{ij}^{best} \leftarrow X_{ij} | f(X_{ij}^{best}) < f(X_{ij}), i = 1, \dots, |S|,$$

3.3. 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. On the basis of this positions, the lists of the best and perspective positions $M = (N_{ij}^b, N_{kj}^g)$ are generated, found at the $(j-1)$ -th iteration. 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 of particles are sent to the vicinity of each best position and c^g are sent to the vicinity of each perspective position. Thus, the positions of all worker particles are determined as follows:

$$\begin{aligned} X_{(i-1)cb+kj} &\leftarrow N_{ij-1}^b + \text{random}(-1, 1) \cdot rad, i = 1, \dots, n^b, \\ k &= 1, \dots, c^b, \\ X_{nb-cb+(i-1)cg+kj} &\leftarrow N_{ij-1}^g + \text{random}(-1, 1) \cdot rad, i = 1, \dots, \\ n^g, k &= 1, \dots, c^g, \end{aligned}$$

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:

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

3.4. If at the j -th iteration a stop-condition is satisfied, then the value X_{ij}^{best} is transmitted to output O , or the transition to iteration 3.2 takes place.

4. 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, coefficient rx defines 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 in order to increase the algorithm efficiency it is necessary to adapt parameters.

3.2.3 Firefly Optimization

Firefly Optimization was proposed by Xin-She Yang in 2010 [21]. This algorithm as all Swarm Intelligence algorithms is based on the particles (fireflies) movement in the search searching space. Let's consider the objective function minimum problem of the following type $f(X)$, where X is a vector of varied parameters which can get the values from some D area. Each particle is specified by the value of X parameter and value of an optimized function $f(X)$. Thus, the particle is a feasible solution of the considered optimization problem.

As the algorithm is based on watching for fly's behavior, each particle is considered to see the "light" from their neighbors, but the brightness of the "light" depends on the distance between particles. For the process of solution finding to be converged to the optimum, each particle in its movement takes into account only those neighbors having a better value of $f(X)$ criterion. But for the algorithm not to degenerate into greedy heuristics, it is necessary to have particles' stochastic movement.

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

1. Set of particles (fire-flies). $S = \{s_1, s_2, \dots, s_{|S|}\}$, $|S|$ is a number of particles. At iteration j the i^{th} particle is specified by the state $s_{ij} = \{X_{ij}\}$, where $X_{ij} = \{x_{1ij}, x_{2ij}, \dots, x_{lij}\}$ is a vector of the varied parameters (particle's position), l is a number of the varied parameters.

2. Means of indirect exchange is vector M is particles' brightness.

$$M = \{f(X_{1j}), f(X_{2j}), \dots, f(X_{|S|j})\}$$

Brightness is determined by the optimality criterion. This vector ensures the indirect experience exchange among particles.

3. Algorithm A describes the steps of the ACO algorithm.

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

$$X_{i1} \leftarrow \text{random}(G(X)), i = 1, \dots, |S|,$$

where $\text{random}(G(X))$ is a vector of equally distributed random variables meeting the restrictions of searching space.

3.2. 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 $\{I, O\}$.

$$m_{ij} \leftarrow f(X_{ij}), i = 1, \dots, |S|$$

$$X_j^{best} \leftarrow X_{ij} | f(X_{ij}) \leq f(X_j^{best})$$

3.3 Particles' movement:

$$X_{ij+1} \leftarrow X_{ij} + v(X_{ij}, X_{kj}) \cdot (X_{ij} - X_{kj}) + \alpha \cdot \text{random}(0, 1) |$$

$$m_{kj} \leq m_{ij}, i, k = 1, \dots, |S|, i \neq k,$$

$$\text{if } G(X_{ij+1}) = 0, X_{ij+1} \leftarrow X_{ij}, i = 1, \dots, |S|,$$

Where $\text{random} \in [0, 1]$, and $G(X)$ is used in this case as the predicate showing if X belongs the area of admissible solutions.

The function $v(X_{ij}, X_{kj})$ defines the attractiveness of k particle for i particle with j algorithm iteration:

$$v(X_{ij}, X_{kj}) \leftarrow \beta \cdot (1 + \gamma \cdot r(X_{ij}, X_{kj}))^{-1}$$

where $r(X_{ij}, X_{kj})$ is Cartesian distance between particles.

3.4. If at the j -th iteration a stop-condition is satisfied, then the value X_j^{best} is transmitted to output O , or the transition to iteration 3.2 takes place.

4. Vector $P = \{\alpha, \beta, \gamma\}$ are coefficients of the algorithm. Coefficient α determines the influence degree of stochastic algorithm nature. Coefficient β sets the degree of attraction between particles with zero distance between them i.e. defines the particle's mutual influence. Coefficient γ controls the dependence of attraction on the distance between particles.

3.3 Application the Swarm Intelligence Algorithm for GC optimal control

For applying SI algorithms, it is necessary to determine the mapping of the particle coordinate (X) in the search space solution to the solutions of the solved task. In this case, the solution is the control actions $A(t)$, as shown in expression (1). Each element of the vector X is bounded from 0 to 1 [15]. The priorities are real numbers from 0.0 to 1.0, so $pr_i = x_i, i = 1, \dots, 12$. The parameters $buy_unload, sale_unload, buy_storage, sale_storage$ also take values from 0.0 to 1.0, so they are mapped in the same way. To set values of $time_1, \dots, time_4$, we use rounded down values of $24x_{17}, \dots, 24x_{20}$.

The FFO algorithm requires comparing each particle to each other, so the number of operations quadratically depends on the number of particles. The PSO and BA have a linear relationship. We reduce the number of FFO particles to equalize the calculation time. At the same time, we increase the number of iterations of the FFO algorithm to equalize the number of calculations of the objective function. As a result, the number of particles is reduced four times, and the number of iterations is increased four times compared to the PSO algorithm and the BA. The parameters of the SI algorithms are given in Table 1.

Table 1. Parameters of the SI algorithms

Alg.	Particles	Iteration	Heuristic coefficients
PSO	200	500	$\alpha_1 = 1.5, \alpha_2 = 1.5,$ $\omega = 0.7, \beta = 0.5$
BA	200	500	$n^s = 60, n^b = 6, n^g = 1,$ $c^b = 20, c^g = 20,$ $rad = 0.01, rx = 0.05$
FFO	50	2000	$\alpha = 0.05, \beta = 1, \gamma = 0.5$

4 Results and Discussion

4.1 Computational Experiment

Computational experiments were carried out while considering the GC of Russky and Popov Islands (GC_R , GC_P , respectively). Table 2 shows the prices used in the simulation.

Table 2. Prices used in the simulation

Power flow	Price, rubel / MWh	Price, \$ / MWh
Wind generation	500	6,67
Power storage discharging	100	1,33
Sale (daily rate)	3200	42,7
Sale (daily rate)	1400	18,67
Buy (daily rate)	2700	36,00
Buy (daily rate)	900	12,00

To evaluate the effectiveness of the rule-based control model built by the Swarm algorithms, we compared them with a base constructed manually by an expert. The main advantage of using SI is an automatic adaptation to the profiles of production and consumption of each GC. Therefore, the expert rules were constructed one time for the general case.

Obviously, in the problem under consideration, control is impossible without power storage. Because without it, GC has to sell electricity at times of excess and buy at times of shortage, and there are no other options. Therefore, control efficiency is limited by the capacity of the power storage. Due to the limitation of capacity and the use of tariff simulation with only two rates (day, night), the reduction in energy consumption or the increase in income (this is the same thing) is not very large. Thus, for results clarification, we did not compare the absolute values of the cost of electricity according to criterion (4), but the benefits that control gives regarding the situation without power storage.

In section 2.1, three cases are considered: the control of a GC of Russky Island, GC of Popova Island, and an integrated system of both GCs. For each situation and each algorithm, modeling was performed on data for two summer months (Fig. 2-4).

4.2 Simulation results

Table 3 shows the results. Each SI algorithm was launched 20 times, and in 17-18 launches out of 20 gave the same result. This result is used as a summary. The results without the use of power storage are shown in rows with the label "No" in the "Algorithm" column, and the results of control using the expert rules are labeled as "Expert". Also, Fig. 7 visualizes the benefits of using the rule-based model optimized by the SI.

It can be seen that the difference varies greatly depending on the profile of production and consumption.

For the GC of Russky Island, there is a situation of electricity shortage, so control does not make a large contribution. For the Popova Island GC, on the contrary, there is an excess of electricity; therefore, it is necessary to determine the best moments for the sale of electricity and the balance between sale and accumulation.

The most interesting situation is when a joint system of two GCs is controlled together. First, in this case, the profile is more complicated (Fig. 4), since there are moments of both excess and shortage of electricity. Secondly, the power storage capacity is two times higher due to the combination of storages of both GC in a single power system. Thirdly, the combined GC has a higher generation and consumption, since, it is evident that all quantitative indicators will be more top.

The results of all applied SI algorithms are very close, even without adjusting the heuristic parameters. It can be explained by the relatively low complexity of the task from the point of view of optimization theory since there are not many control options.

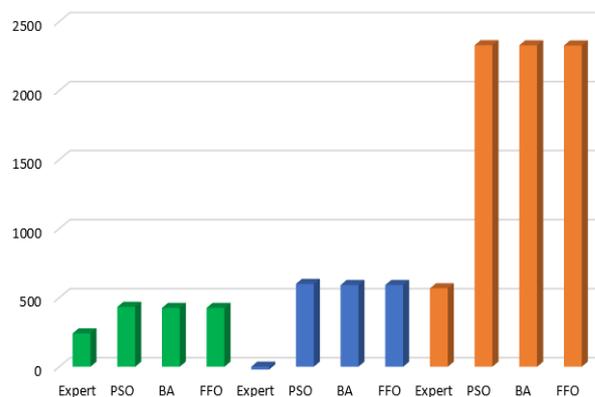


Fig. 7. Algorithms' results. The histogram shows the monthly profit (\$) of power storage usage with the different algorithms of optimize control rule-based model. Left 4 bars shows results for GC_R , central 4 bars – for GC_P , right 4 bars – for GC_{R+P} .

Table 3. Algorithms' results

Case	Algorithm	Cost, thousands \$	Monthly Profit, thousands \$
GC_R	No	580.3	-
GC_R	Expert	579.8	0.241
GC_R	PSO	579.4	0.43
GC_R	BA	579.5	0.43
GC_R	FFO	579.5	0.43
GC_P	No	-41.6	-
GC_P	Expert	-41.5	0
GC_P	PSO	-42.8	0.6
GC_P	BA	-42,8	0.59
GC_P	FFO	-42,8	0.59
GC_{R+P}	No	452,0	-
GC_{R+P}	Expert	450,4	0.57
GC_{R+P}	PSO	447,3	2.33
GC_{R+P}	BA	447,3	2.33
GC_{R+P}	FFO	447,3	2.33

5 Conclusion

In this research, we have applied Swarm Intelligence algorithms to optimize the heuristic-rule-based control model (optimize priorities of rules and numerical values of the model coefficients) for optimal control of generating consumers with wind power plants. The computer simulation showed that SI allowed to increase the profit of power storage usage 1.8–4.1 times compared with the rules build by experts. The simulation results confirmed that it is appropriate to apply the SI algorithms to increase accuracy of heuristic-rule-based model, and perform adaptation to a given generating consumer.

The proposed control model allows to get the robust control in a particular situation, which can be easily transferred to other climatic conditions and GC features. For future work, we plan: firstly, to complicate a GC model and a model of GCs interaction; secondly, to apply the Q-learning method for the optimal control of GC; thirdly, make comparisons with existing methods using larger dataset.

This work is supported the Novosibirsk State Technical University Development Program through the Project C20-20.

References

1. C.W. Gellings. *The Smart Grid: enabling energy efficiency and demand response*. (Lilburn, CA: Fairmont Press, 2009).
2. V.Z. Manusov, N. Khasanzoda, P.V. Matrenin. *Application of artificial intelligence methods to energy management in Smart Grids* (Novosibirsk, 2019) [In Russian]
3. X. Fang, S. Misra, G. Xue, D. Yang. Managing smart grid information in the cloud: Opportunities model and applications. *IEEE Netw.*, **26.4**, 32-38 (2012).
4. R. Zafar, et. al. Prosumer based energy management and sharing in smart grid. *Renewable and Sustainable Energy Reviews*, **82.1**, pp. 1675-1684 (2018)
5. A.G. Azar, et. al. A Non-Cooperative Framework for Coordinating a Neighborhood of Distributed Prosumers. *IEEE Transactions on Industrial Informatics*, **15.5**, pp. 2523-2534 (2019)
6. L. Ma, N. Liu, J. Zhang, L. Wang. Real-Time Rolling Horizon Energy Management for the Energy-Hub-Coordinated Prosumer Community From a Cooperative Perspective. *IEEE Transactions on Power Systems*, **34.2**, pp. 1227-1242 (2019)
7. A.C. Luna, N.L. Diaz, M. Graells, J.C. Vasquez, J.M. Guerrero. Cooperative energy management for a cluster of households prosumers. *IEEE Transactions on Consumer Electronics*, **62.3**, pp. 235-242 (2016)
8. H. Mortaji, S. Siew, M. Moghavvemi, H. Almurib. Load Shedding and Smart-Direct Load Control Using Internet of Things in Smart Grid Demand Response Management. *IEEE Transactions on Industry Applications*, **53.6**, pp. 5155-5163 (2017)
9. S.R. Etesami, W. Saad, N.B. Mandayam, H.V. Poor. Stochastic Games for the Smart Grid Energy Management with Prospect Prosumers. *IEEE Transactions on Automatic Control*, **63.8**, pp. 2327-2342 (2018)
10. G. El Rahi, et. al. Managing Price Uncertainty in Prosumer-Centric Energy Trading: A Prospect-Theoretic Stackelberg Game Approach. *Transactions on Smart Grid*, **10.1**, pp. 702-713 (2019)
11. N. Liu, X. Yu, C. Wang, J. Wang. Energy Sharing Management for Microgrids with PV Prosumers: A Stackelberg Game Approach. *IEEE Transactions on Industrial Informatics*, **13.3**, pp. 1088-1098 (2017)
12. Y. del Valle, et al. Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems. *IEEE Transactions on Evolutionary Computation*, **12.2**, pp. 171-195 (2008)
13. V.Z. Manusov, P.V. Matrenin, L.S. Atabaeva. Firefly Algorithm to Optimal Distribution of Reactive Power Compensation Units. *International Journal of Electrical and Computer Engineering*, **8.3**, pp. 1758-1765 (2018)
14. V.Z. Manusov, et al Optimization of Power Distribution Networks in Megacities. *IOP Conference Series: Earth and Environmental Science. International Conference on Sustainable Cities*, **72**, id 012019 (2017)
15. V.Z. Manusov, P.V. Matrenin, S.E. Kokin. 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*, **12.1**, pp.101-112 (2017)
16. N. Khasanzoda. *Optimization of power consumption modes in intelligent networks with a two-way flow of energy using artificial intelligence methods* (Novosibisk, 2019) [In Russian]
17. L.A. Rastrigin, *Modern principles of management of complex objects* (Moskov, 1980) [In Russian]
18. J. Kennedy, R. Eberhart. Particle swarm optimization. *IEEE International Conference on Neural Networks*, Perth, WA, Australia. pp. 1942–1948 (1995)
19. R.C. Eberhart, Y. Shi. Particle swarm optimization: developments, applications and resources. *Congress on Evolutionary Computation*; Seoul, South Korea. pp. 81-86 (2001)
20. D.T. Pham, et. al. *The bees algorithm – a novel tool for complex optimisation problems* (Cardiff, UK, 2005)
21. X. Yang. Firefly algorithm, Stochastic Test Function and Design Optimization. *International Journal of Bio-Inspired Computation*. **2.2**, pp. 78-84 (2010)