

# New method of stochastic optimization for dynamic optimal power flow

Alexander Domyshev<sup>1\*</sup>

<sup>1</sup>Melentiev Energy Systems Institute of SB RAS, Irkutsk, Russia

**Abstract.** An universal algorithm for stochastic optimization is proposed. This algorithm is effective for dynamic optimization of process changing in time with taking into account the time-dependent cost of actions. Proposed algorithm is tested on the model of quite big power system and proved to be effective.

## 1 Introduction

The article proposes a methodology for modeling and dynamic optimization of electrical networks with stochastic elements. This task is relevant for the automatic and automated control of normal electric power system (EPS) states.

The initial data for the optimization of the regime is the forecast of changes in the parameters of the regime for a given time. The forecast horizon is divided into separate time steps at regular intervals. On every time step there is the minimum required set of input data for calculating the power flow.

Optimization aims for such a network from the point of view of a network company can be:

- minimization of losses in the network,
- restoration of the required voltage levels (in case of their violation),
- optimization of electricity generation cost.

The task of dynamic optimization is to minimize the total objective function over the entire forecast time horizon by choosing the compound and time of control actions  $x_t$  for each time point in the forecast range:

$$\min \sum_{t=0}^T f_t(x_t)$$

Optimization in the general case is a multi-objective optimization that can be solved by the weighted sum optimization method [1, 2], when individual objective functions are summed into one using appropriate weights.

$$f_c(x) = \sum_{i=1}^n w_i f_i(x) \quad \left| \quad g(x) \leq 0 \right.$$

where  $f_i(x)$  is the objective function according to one of the considered criteria;

$w_i$  – weight coefficient corresponding to the objective function subtask;

$x$  is a vector of control parameters;

$g(x)$  – constraints, both on the control parameters themselves, and functional constraints.

Unlike static optimization of one state, in the dynamic optimization problem, it is important to take into account the “cost” of control actions, which depends not only on the system state vector, but also on time.

The cost of managing one or another equipment depends on factors such as:

- residual resource of equipment;
- priority of using control actions;
- the minimum allowable time between switchings with the same device.

The optimization problem, taking into account the cost of the impact, is written as:

$$\min \sum_{t=1}^T f_t(X_t, t) = \sum_{t=1}^T \left( f_{d_t}(X_t) + \sum_{i=1}^c f_{c_i}(x_{t_i}, t) \right)$$

where  $X_t$  – control actions available at time  $t$ ;  $f_{d_t}$  – function of static optimization of each mode for time  $t$ ;  $f_{c_i}$  is a monotonically decreasing function of the cost of the control action  $x_{t_i}$ , depending on the time of actions that were performed before time  $t$ . An approximate form of the penalty function is presented by expression (1). The meaning of this expression is that after applying the control action, its repeated application for some time should be blocked by a high value of the penalty function (Fig. 1). After some time, the cost of exposure decreases to a constant value  $c_i$ .

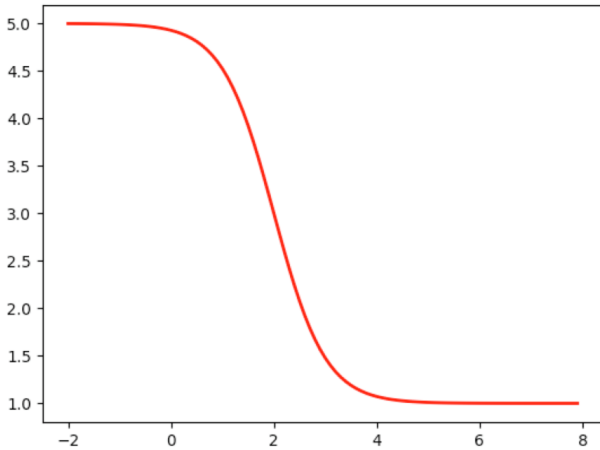
$$f_{c_i} = k_{c_i}(th(k_{d_i}(t_{p_i} - t + \Delta t_b)) + 1) + c_i \quad (1)$$

where  $t_{p_i}$  – is the time of the previous application of this control action.

$k_{c_i} > 0, k_{d_i} > 0, c_i \geq 0, \Delta t_b > 0$  are some scaling and shifting constants.

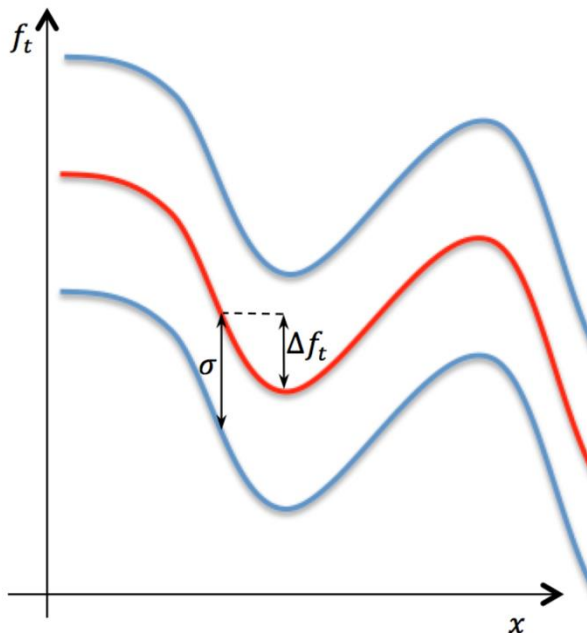
The penalty function formula (1) makes it possible to ensure the blocking of the reuse of this action while ensuring the smoothness of the objective function after applying this action.

\* Corresponding author: [domyshev@isem.irk.ru](mailto:domyshev@isem.irk.ru)



**Fig. 1.** An example of the cost function of control.

The presence in the objective function of the dependence on the time of application of the previous control actions makes the optimization process not a Markov process. Since the “cost” of the impacts and the dependence of the “cost” on time for different devices are different (for example, the tap changer of transformers can be switched rarely, but it can be controlled with the help of control shunt reactor quite often), it will not be possible to tune out such effects and return the Markov properties to the optimization process.



**Fig. 2.** Impact of the variance of the objective function on the optimization process.

An additional complexity is introduced by the need to take into account the stochasticity of the initial data. The value of the variance of the objective function  $\sigma$  can be greater than the improvement in the value of the objective function in the process of optimization  $\Delta f_t$  (Fig. 2). In the limiting case, if the value of the uncertainty of the objective function is such that it can

almost equally likely take any value from the permissible range, then it makes no sense to optimize such a regime.

## 2 Statement of the problem of dynamic optimization of UPS states

The problem of automatic optimal control of the electrical regime can be rather simplified, but it is quite adequately represented by the following model:

$$\min \sum_{t=1}^T \left( f_t(S_t, Y_t) + \sum_{i=1}^{n-1} \sum_{j=i+1}^n f_{cij}(t) \right), \quad (2)$$

where  $S_t$  is the vector of power injections for the  $t$  time;  $Y_t$  is the matrix of the admittances of the branches, the elements of which  $y_{tij}$  are control actions;

$f_{cij}$  is a monotonically decreasing function of the cost of the control action  $y_{tij}$ , depending on the actions that were performed before time  $t$ .

The first stage of dynamic optimization is proposed to estimate the lower bound of the optimal solution and select from the entire set of control actions those that affect the optimal solution for the considered time range.

To obtain the lower bound for the global optimization solution, a static optimization problem is solved for each time point. In this case, the time-dependent component of the penalty is excluded from the objective function. The optimization problem for each point in time is written as:

$$\min f_{it}(S_t, X_t) = f_{dt}(S_t, X_t) + \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij}, \quad (3)$$

where  $c_{ij}$  is a constant component of the cost of the control action,  $S_t$  is the vector of free mode parameters sufficient for calculating the steady state (voltage power injection in balancing nodes),  $X_t$  are control actions available at time  $t$ .

As a result of the calculation for each moment in time  $f_{it} < f_t$ . The total objective function for the entire time range  $[1...T]$  will also be less than when taking into account the cost components of time-dependent impacts.

The problem of using existing stochastic optimization algorithms, such as the simulated annealing method for solving the problem of dynamic optimization of modes, is the search for an optimum over the entire space of optimization parameters, one of the measurements of which is time.

## 3 The proposed stochastic optimization algorithm

The lower bound of the objective function (2) for each time point is determined by the results of static optimization. For most moments of time in the process of optimal control, the term in the objective function, which depends on time (Fig. 1) is close to zero. To use this feature of the dynamic optimal power flow, a special algorithm for stochastic optimization of the dynamic process was developed, This algorithm efficiently finds a

set of optimal actions taking into account the time-dependent component of the objective function. The algorithm is based on the principles of optimization by the particle swarm method [3].

The classical Particle Swarm Optimization algorithm is as follows. A swarm of particles is created. Each particle is an agent represented by coordinates in the space of control parameters and a velocity vector in this space. Each particle remembers the coordinates of the best solution found to it, and also knows the coordinates of the best solution found by the whole swarm of particles.

Initialization of the initial coordinates and velocities of the particles is performed randomly, usually uniformly over the entire volume of the space of control actions, limited by the permissible limits of parameter variation.

During the optimization process, the particle speed is changed according to the following algorithm:

$$v_i(t+1) = \omega v_i(t) + c_l r_1 (y_i(t) - x_i(t)) + c_g r_2 (g(t) - x_i(t)),$$

where  $v_i(t)$  is the speed of the  $i$ -th particle at time  $t$ ,

$\omega$  - coefficient of inertia,

$c_l$  - coefficient of movement to the local optimum,

$c_g$  - coefficient of movement to the global optimum,

$y_i(t)$  - coordinates of the local optimum found by the time  $t$ ,

$g_i(t)$  - coordinates of the global optimum found by time  $t$ ,

$r_1, r_2$  - random values obtained at each optimization iteration.

The coordinate of the particle in the solution space changes as follows:

$$x_i(t+1) = x_i(t) + v_i(t)$$

A new algorithm based on the particle swarm algorithm is proposed, which solves the problem of dynamic optimal power flow with a large number of control actions. The working title of the algorithm is "River Stone PSO". The meaning of this name will be clear based on its further description. In the optimization process, we move along the optimal trajectory determined by the results of static optimization. The objective function of dynamic optimization (3) can be rewritten as:

$$\min \sum_{t=1}^T (f_0(t) + \xi(t)), \quad (4)$$

where  $f_0(t)$  is the component of the objective function at the moment of time  $t$ , independent of the effects performed in the past moments of time ( $t-i$ ) and obtained as a result of static optimization of the electrical regime for each moment of time;

$\xi(t)$  - component of the objective function depending on the impacts performed in the past moments of time (1).

If the value  $\xi(t)$  at the optimization step exceeds a certain threshold value  $\xi(t) > \varepsilon$ , then a range of times (slices) for which this condition is satisfied is selected and for this range, dynamic optimization is performed by the proposed algorithm. In this case, the upper range in

the total objective function is the value of the time index  $e$  at which the value of  $\xi(t)$  becomes less than  $\varepsilon$ .

$$\min \sum_{t=1}^e (f_0(t) + \xi(t))$$

After a new range of time points is selected, at which dynamic optimization is required, a swarm of particles is created. Unlike the classical particle swarm algorithm, particles are not generated over the entire volume of the solution space, but are created at the current point determined by the results of static optimization at which the excess  $\xi(t) > \varepsilon$  first occurred.

The initial velocity of each particle is set randomly, but so that in the measurement of the corresponding time the velocity component is negative, that is, the initial velocity is directed backward in time.

For the directions of the search space for solutions corresponding to the control actions, the initial velocity is given as:

$$v_{0_i} \sim k_v [-1; 1]$$

For the direction of the solution search space corresponding to the time, the initial velocity is given as:

$$v_{0_i} \sim -k_v [0; 1]$$

In the algorithm for changing the speed, one more point of attraction is added for each particle. In addition to the optimal value found by the particle itself and the global optimum found by the entire swarm of particles, a point  $e$  is added corresponding to the end of the range  $\xi(t) > \varepsilon$  and the control actions obtained as a result of static optimization for the moment of time  $e$ :

$$v_i(t+1) = \omega v_i(t) + c_l r_1 (y_i(t) - x_i(t)) + c_g r_2 (g(t) - x_i(t)) + c_e r_3 (y_e - x_i(t))$$

where  $y_e$  is the coordinate corresponding to the end of the interval  $\xi(t) > \varepsilon$ ,

$c_e$  - coefficient of movement towards the end of the interval,

$r_1, r_2, r_3$  - random values obtained at each optimization iteration.

An illustration of the proposed algorithm is shown in Fig. 3. Moving along the optimal trajectory, we stumble upon a high value of  $\xi(t)$ , "rebound" back from it in time and find the optimal trajectory enveloping high values of the objective function, like a river bends around a stone lying in its bed. This is where the name of the algorithm comes from.

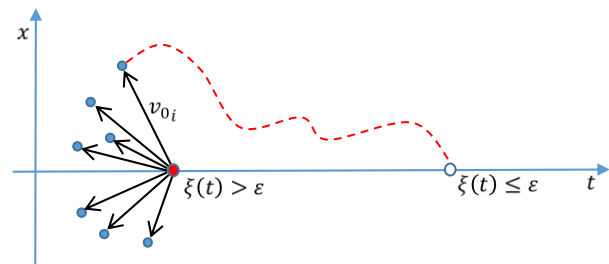


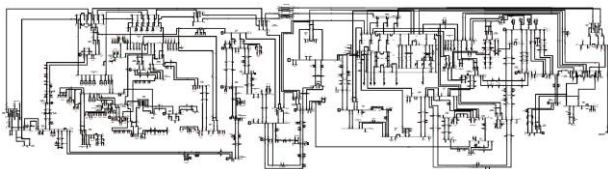
Fig. 3. Illustration of the RS PSO algorithm.

The proposed algorithm can be used not only for dynamic optimal power flow, but also for optimal

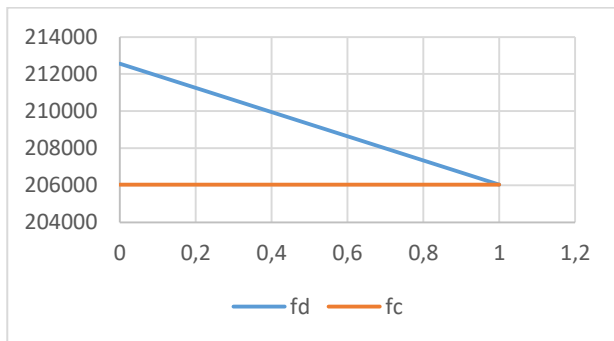
control of any, where the cost of the control action depends on the past time of its application.

As an example for dynamic optimization, a model of a real power system (Fig. 4), prepared in the ANARES complex [4], was chosen. The number of nodes in the models nodes / branches of this electrical network is 1248 and branches - 1481.

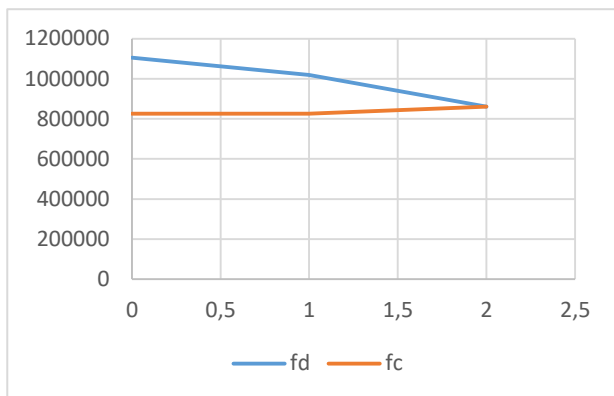
For this power system, there are archived data presented in the form of a set of electrical modes with a frequency of 30 minutes. The depth of the archive is 1 month. However, the archive data optimization algorithm itself is sufficient to check. In the industrial application of this algorithm, optimization is carried out based on the power flow forecast.



**Fig. 4.** Considered electrical grid.



**Fig. 5.** Changing the objective function during dynamic optimization RS PSO for 1st range.

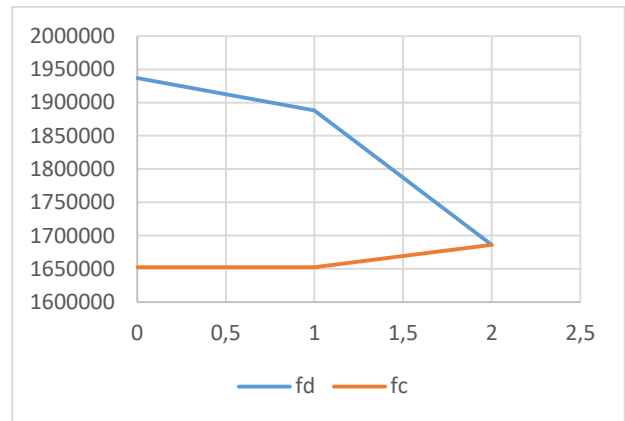


**Fig. 6.** Changing the objective function during dynamic optimization RS PSO for 2nd range.

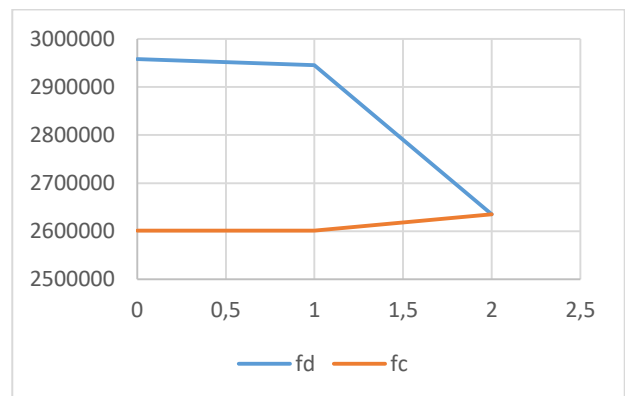
When considering a month time range in dynamic optimal power flow, 6 ranges were identified, on which dynamic optimization needed. For each range of each, no more than 4 iterations were required with 10 generated particles for each control action (the total number of particles is 730). The results of changing the objective function are presented in the figures. 5 – 11.

On the graphs,  $fc$  denotes the total value of the static part of the problem with the current set of control actions,  $fd$  denotes the total value of the function taking into account the time-dependent component.

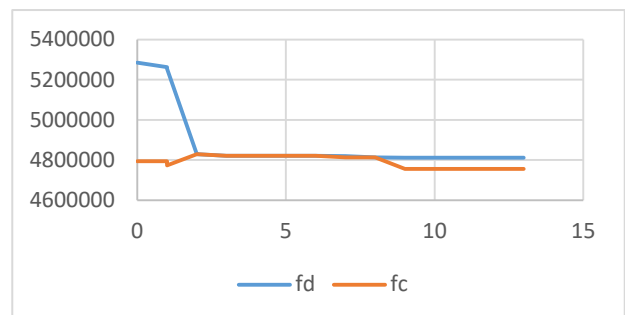
The graphs show that in some cases the static part of the objective function increases, which is logical. It can be seen that the value is either strictly or significantly close to the original minimum boundary of the objective function.



**Fig. 7.** Changing the objective function during dynamic optimization RS PSO for 3rd range.



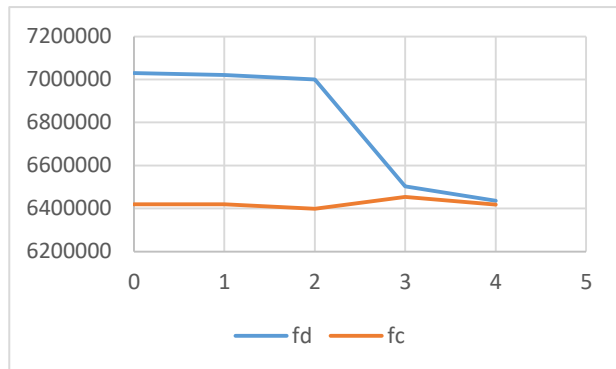
**Fig. 8.** Changing the objective function during dynamic optimization RS PSO for 4th range.



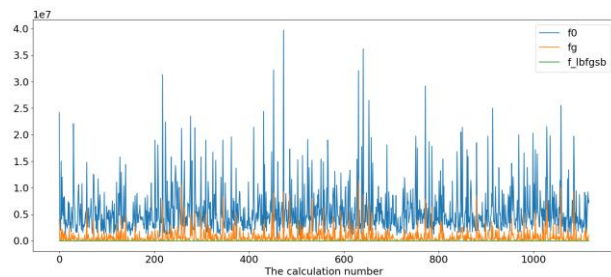
**Fig. 9.** Changing the objective function during dynamic optimization RS PSO for 5th range.

For comparison, an adaptive version of the ultrafast annealing simulation method was carried out. A series of calculations with random initial deviations of the state parameters showed that in some cases during the

operation of the dynamic optimization algorithm the resulting power flow deviates from the optimal one. In addition, the speed of calculation by the RS-PSO method significantly exceeds the speed of other stochastic algorithms.



**Fig. 10.** Changing the objective function during dynamic optimization RS PSO for 6th range.



**Fig. 11.** Optimization results on the EPS of the Irkutsk region with specified optimization constraints

### 3 Conclusions

An universal algorithm for stochastic optimization of a dynamic process in time with taking into account the time-dependent cost of actions is proposed.

The proposed dynamic optimization algorithm has been tested on the model of a real power system.

The proposed methodology for modeling and optimization of distribution networks with stochastic elements is implemented in the form of embedded software and can be used in automatic optimal and emergency control systems.

### References

1. T.W. Athan, P.Y. Papalambros A note on weighted criteria methods for compromise solutions in multi-objective optimization //Engineering optimization. – 1996. – **27** (2) –155-176.
2. E.N. Gerasimov, V.N. Repko Multicriterial optimization, International Applied Mechanics, 1978, **14**(11), 1179-1184.
3. S. Sengupta, S. Basak, R.A. Peters Particle Swarm Optimization: A survey of historical and recent developments with hybridization perspectives

//Machine Learning and Knowledge Extraction. – 2019. – **1** (1) –157-191.

4. ANARES - software for power system modelling [Electronic resource] – Access: <http://anares.ru/software>