Innovative Lifelong e-Learning for Professional Engineers (e-ProfEng)

586391-EPP-1-2017-1-SE-EPPKA2-CBHE-JP

Training in Electrical Engineering Discipline

Modeling and Simulation in Electrical Engineering

Exercises for Topic 3

Theory and implementation of metaheuristic optimization methods for optimizations in the distribution power system

Authors:

Marinko Barukčić, P9 FERIT

Contents

Exercise 1: Optimization of the problem with an analytically objective function

Task: Find optimal powers of the generating units such that power (energy) production cost per hour is as low as possible (unit commitment problem).

Problem description: The electric power system consists of four generating units supplying the loads with energy. Value of total load is constant in time. Each source produces power (energy) at a price described by the production cost function. The optimal amount of each source needs to be found by the metaheuristic optimization procedure.

System data:

The production cost function is given by:

$$C_i = a_i P_i^2 + b_i P_i + c_i \quad i \in (0,1,2,3) \tag{3.5}$$

The cost coefficients values in (3.5) are given in Table 3.3.

Table 3.3: Cost coefficients for each production unit

Coefficient\Unit	0	1	2	3
a [EUR/kW^2]	3e-4	2e-4	8e-4	1e-4
b [EUR/kW]	0.02	0.015	0.024	0.018
c [EUR]	7	5	4	7

Power limit per production unit: 1000 kW

Amount of total system load is P_{load} = 2500 kW.

Specific tasks to solve the optimization problem by using the metaheuristic optimization procedure:

- Formulate the optimization problem (expressions for the objective function and constraints)
- Coding decision variables in the method individual
- Defining ranges (limits) of the decision variables-box constraints
- Coding the problem in the specific optimization tool (SciPy DE).

Guidance for problem solving:

The objective function: summation of cost functions of each unit

The constraint: system power balance equation – equality constraint, unit production range – box constraints

Syntax of ScyPy Differential Evolution function: https://docs.scipy.org/doc/scipy-0.17.0/reference/generated/scipy.optimize.differential_evolution.html

Solution: jupyter notebook ***Optim01.ipynb*** in additional materials, OF = 402.1 EUR, P1 = 522.68 kW, P2 = 786.92 kW, P3 = 190.53 kW, P4 = 1000 kW.

Self exercise 1:

- All units produce in range 300 – 1000 kW (all other data are same as in the basic problem), *solution: OF = 412.51 EUR, P1 = 477.24 kW, P2 = 722.96 kW, P3 = 300 kW, P4 = 1000 kW.*

- Total load is 500 kW (all other data are same as in the basic problem), *solution: OF = 44.63 EUR, P1 = 80.86 kW, P2 = 135.83 kW, P3 = 29.1 kW, P4 = 254.2 kW.*

Exercise 2: Coding distribution networks in OpenDSS

Task: For given example of the distribution network feeder in Figure 3.6, make a model in OpenDSS simulation tool. Data of the network are given in Table 3.4. Simulate a given network and show total losses and phase voltages in all nodes. Loads are modeled as constant power load model.

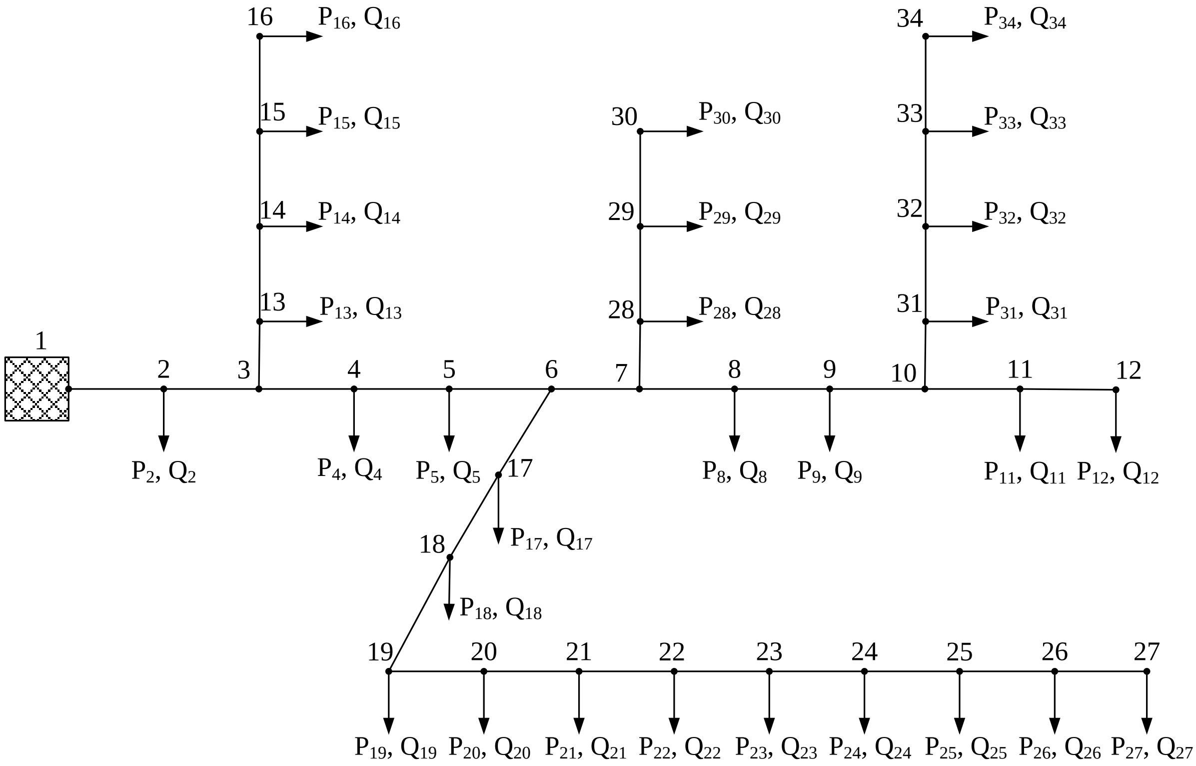

Figure 3.6: 11 kV Distribution network

Table 3.4: Data of the network from Figure 3.6.

Line	Line impedance		Node	Load power	
	R [Ω]	X_L [Ω]		P [kW]	Q [kvar]
1 – 2	0,117	0,048	2	230,00	142,50
2 – 3	0,107	0,044	3	0,00	0,00
3 – 4	0,164	0,046	4	230,00	142,50
4 – 5	0,150	0,042	5	230,00	142,50
5 – 6	0,150	0,042	6	0,00	0,00
6 – 7	0,314	0,054	7	0,00	0,00
7 – 8	0,210	0,036	8	230,00	142,50
8 – 9	0,314	0,054	9	230,00	142,50
9 – 10	0,210	0,036	10	0,00	0,00
10 – 11	0,131	0,023	11	230,00	142,50
11 – 12	0,105	0,018	12	137,00	84,00
3 – 13	0,157	0,027	13	72,00	45,00

13 – 14	0,210	0,036	14	72,00	45,00
14 – 15	0,105	0,018	15	72,00	45,00
15 – 16	0,052	0,009	16	13,50	7,50
6 – 17	0,179	0,050	17	230,00	142,50
17 – 18	0,164	0,046	18	230,00	142,50
18 – 19	0,213	0,047	19	230,00	142,50
19 – 20	0,194	0,043	20	230,00	142,50
20 – 21	0,194	0,043	21	230,00	142,50
21 – 22	0,262	0,045	22	230,00	142,50
22 – 23	0,262	0,045	23	230,00	142,50
23 – 24	0,314	0,054	24	230,00	142,50
24 – 25	0,210	0,036	25	230,00	142,50
25 – 26	0,131	0,023	26	230,00	142,50
26 – 27	0,105	0,018	27	137,00	85,00
7 – 28	0,157	0,027	28	75,00	48,00
28 – 29	0,157	0,027	29	75,00	48,00
29 – 30	0,157	0,027	30	75,00	48,00
10 – 31	0,157	0,027	31	57,00	34,50
31 – 32	0,210	0,036	32	57,00	34,50
32 – 33	0,157	0,027	33	57,00	34,50
33 – 34	0,105	0,018	34	57,00	34,50

Specific tasks to solve the optimization problem by using the metaheuristic optimization procedure:

- Formulate the optimization problem (expressions for the objective function and constraints)
- Coding decision variables in the method individual

Solution: *OpenDSS script in additional materials, P_{loss} = 217.5 kW, V_{min} = 0.9399 p.u. in node b27*.

Self-exercise 2:

- Add two distributed generation (DG) units of sizes P1 = 500 kW, Q1 = 150 kvar, P2 = 800 kW, Q2 = 400 kvar in network nodes 9 and 24. Compare total network losses to case without DGs. *solution: OpenDSS script in additional materials, P_{loss} = 99.3 kW, V_{min} = 0.9631 p.u. in node b27.*
- Add two distributed generation (DG) units of sizes P1 = 500 kW, Q1 = 150 kvar, P2 = 800 kW, Q2 = 400 kvar in network nodes 2 and 16. Compare total network losses to the case without DGs and DGs installed in nodes 9 and 24. *solution: OpenDSS script in additional materials, P_{loss} = 198.2 kW, V_{min} = 0.9427 p.u. in node b27.*

Exercise 3: Optimal allocation of distributed generation (DG) in the distribution network

Task 1: Using OpenDSS – SciPy DE co-simulation find the optimal allocation of DG in distribution network from Figure 3.6 (network data in Table 3.4). DG power limits are PDG ∈ (0, 5000) kW, QDG ∈ (-2000, 2000) kvar. Use the OpenDSS script generated in Exercise 2. All modifications in the OpenDSS script do only from Python except adding DG unit (add it in OpenDSS script in any network node). Nodal voltage values required to be within the limits of 0.9 p.u. ≤ V ≤ 1.1 p.u. The problem objective is the minimization of total active power losses. (Tip: repeat the optimization procedure more time changing parameters of DE each time to find their values guarantee convergence of the solution near to the global optimum).

Task 2: As Task 1 but with loads changing in time according to the dynamic given in Figure 3.8. Production of DG is required to be constant over time.

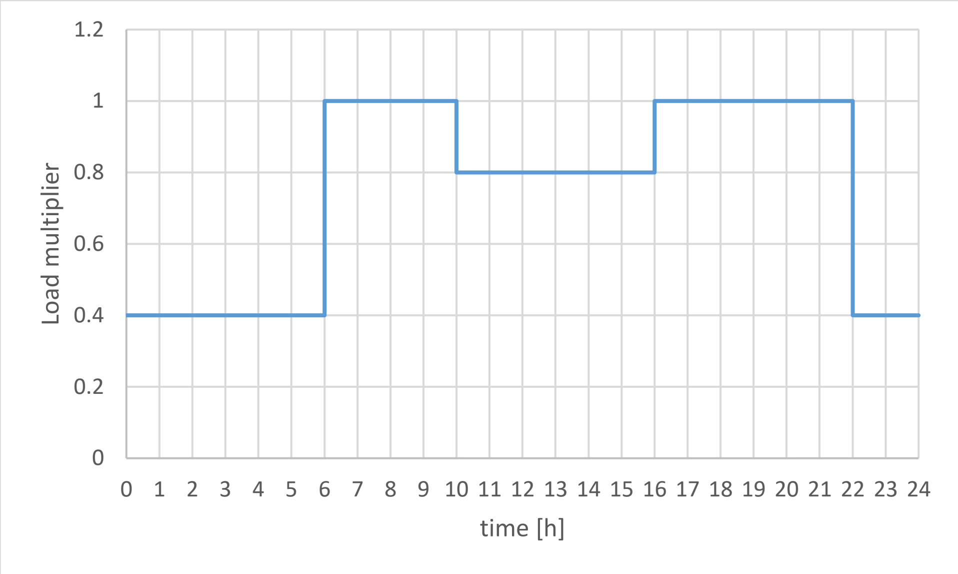

Figure 3.8: Load shape in Task 2 (Exercise 3)

Specific tasks to solve the optimization problem by using co-simulation approach:

- Set COM interface to OpenDSS
- Coding decision variables as DE individual
- Add DG unit to prepared OpenDSS script through COM interface from Python
- Formulate the optimization problem in black-box approach – function in Python including an interface to OpenDSS
- Formulate and code constraints – simultaneously with coding the objective function
- Defining ranges (limits) of the decision variables
- Execute optimization and show results

Solution:

Add DG unit in OpenDSS script prepared in Exercise 2:

New Generator.DG1 bus1=LOCd …

DE individual can be formulated as ind = [DG_L, DG_P, DG_Q] with decision variables of location, active and reactive powers of DG. So, this is a three-dimensional problem - 3 decision variables.

Because SciPy DE works with continuous variables the first decision variable need to be coded as integer values represents the network node (location of the DG):

DGL = round(ind[0])

Location represents position in bus list:

BusN = [bus 1, bus2, … busN]

It will be decoded to OpenDSS syntax as:

LOCd = BusN[DGL]

dssText.Command = 'Edit generator.DG1' + 'bus1='+LOCd+' kw='+str(ind[1])+' Kvar='+str(ind[2])

Solution of Task 1: Complete solution in form of OpenDSS and Python scripts can be found in additional materials, and optimal allocation is: *bus b21, PDG = 2937 kW, QDG = 1800 kvar,* P_{LOSS} *= 46.93 kW,* V_{min} *= 0.9823 p.u. ,* V_{max} *= 0.9993 p.u.*

Solution of Task 2: Complete solution in form of OpenDSS and Python scripts can be found in additional materials, and optimal allocation is: *bus b21, PDG = 2217.47 kW, QDG = 1358.77 kvar,* P_{LOSS} *= [27.25349751651501, 30.15595202862467, 56.55781826046511] kW,* E_{LOSS} *= [218.0279801321201, 180.93571217174804, 565.5781826046511] kWh,* E_{LOSST} *= 964.54 kWh,* V_{min} *= [1.000, 0.9849, 0.9770] p.u. ,* V_{max} *= [1.016, 0.9994, 0.99907]p.u.*

Self-exercise 3:

Solve Task 1 from exercise 3 with possible installation of 5 DG units. What is the optimal number of DGs?. *Solution: OpenDSS script and PY scripts in additional materials, buses ['b21', 'b17', 'b25', 'b9', 'b30'], PDG = [899.52, 916.98, 954.98, 796.53, 492.49] kW, QDG = [558.18, 557.77, 588.11, 495.09, 307.61] kvar, PDGt = 4060.5 kW,* P_{LOSS} *= 3.71 kW,* V_{min} *= 0.9979 p.u. ,* V_{max} *= 0.9998 p.u.*

Exercise 4: Optimal allocation of distributed generation (DG) in the distribution network for two objective optimization problem

Task: Using OpenDSS – DEAP co-simulation find the optimal allocation of DG in distribution network from Figure 3.6 (network data in Table 3.4). DG power limits are PDG ∈ (0, 5000) kW, QDG ∈ (-2000, 2000) kvar. Use the OpenDSS script generated in Exercise 2. All modifications in the OpenDSS script do only from Python except adding DG unit (add it in OpenDSS script in any network node). Nodal voltage values required to be within limits of 0.9 p.u. ≤ V ≤ 1.1 p.u. The problem objectives are the minimization of total active power losses and minimization of active DG power. Loads are constant over time. (Tip: repeat the optimization procedure more time changing parameters of EA each time to find their values guarantee convergence of the solution near to the global optimum).

Specific tasks to solve the optimization problem by using co-simulation approach:

- Set COM interface to OpenDSS
- Coding decision variables as DEAP EA individual
- Add DG unit to prepared OpenDSS script through COM interface from Python
- Formulate the optimization problem in black-box approach – function in Python including an interface to OpenDSS
- Formulate and code constraints – simultaneously with coding the objective function
- Defining ranges (limits) of the decision variables
- Execute optimization by using DEAP and show results

Solution:

Add DG unit in OpenDSS script prepared in Exercise 2:

New Generator.DG1 bus1=LOCd …

EA individual can be formulated as ind = [DG_L, DG_P, DG_Q] with decision variables of location, active and reactive powers of DG. So, this is a three-dimensional problem - 3 decision variables.

DG location represents position in bus list:

LOC = [bus 1, bus2, … busN]

It will be decoded to OpenDSS syntax as:

LOCd = LOC[DGL]

dssText.Command = 'Edit generator.DG1' + ‘bus1='+LOCd+' kw='+str(ind[1])+' Kvar='+str(ind[2])

Parts of EA should be coded for using DEAP based on the guidance given in previous subsection.

Complete solution in form of OpenDSS and Python scripts can be found in additional materials, and optimal allocation is: *optimal location in Pareto set are in busses b21, b22 and b23, P_{DG} is in range 1.21-2886.4 kW, Q_{DG} is in range 1522.97-1787.62 kvar, Ploss is in range 46.97-171.41 kW.*

Homework:

- Solve Task from exercise 4 with the possible installation of 10 DG units. What are optimal allocations of DGs for edges of the Pareto front (minimal losses and maximal voltage profile flatness)?