PA evaluations

Kaifeng Yang, Furong Ye

Universiteit Leiden
The Netherlands

General Info. – (1)

- Deadline:
  - **Submission: Friday November 9, 23:59**
  - Submit your report/codes on Blackboard

- Penalty: -0.5/week

- Bonus: 0.5/group for the highest PA grades

- Can work with a group (max. 2 persons)
  - Register your team on Blackboard

General Info. – (2)

- How to evaluate your PA? Average grade over all the sections.
  - Following the guidelines (sec.1)
  - Algorithm level (sec.2)
  - Code Reproducibility (sec.3)
  - Experimental Results (sec.4)
  - Presentation (sec.5)
  - Code (sec.6)
  - Overall impression (sec.7)
- Formula:
  - $\left(\frac{1}{7}\sum_{i=1}^{7} sec.i\right) + penalty + bonus$
- Other:
  - **Plagiarism check**: if report copies more than 30%, PA grade is 0

General Info. – (3)

- Implementation of:
  - MC algorithm
  - GA algorithm

Sec.1 – Guidelines

- Format
  - Two and only two MATLAB files (no toolbox)
  - Named by: *studentnumber1_studentnumber2_ga.m,* eg. s12345_s12346_ga.m
  - *studentnumber1_studentnumber2_mc.m, eg. s12345_s12346_mc.m*
  - Report: ≥ 4 pages,
  - NO CODES in your report
  - Delivery the hard copy of your report to Furong Ye (room # 163)
- Report structure:
  - Introduction
  - Problem Description
  - Implementation (GA and MC)
  - Experiments
  - Discussion and Conclusion
- See PA1_introduction.pdf for details

Sec.2 – Algorithm level

- Operators
  - Crossover
  - Mutation
  - Selection
  - etc.

Sec.3 – Code Reproducibility

- Whether a reader can reproduce your code just from your report

Sec.4 – Experimental Results

- Not evaluated by the results in your report

- Evaluated by the rank of your results, compared to your classmates.
  - Run 10 times
  - Number of function evaluation: 10,000
  - Evaluated by mean f and corresponding std.

Sec.5 – Presentation

- Explain/Illustrate:
  - Problem description
  - Algorithm
  - Visualize your experimental results
  - etc.

Sec.6 – Code

- Neat or messy

- Description of the function?
  - Inputs
  - Outputs
  - Example etc.

Sec.7 – Overall impression

- Overall impression

Any questions for PA1?

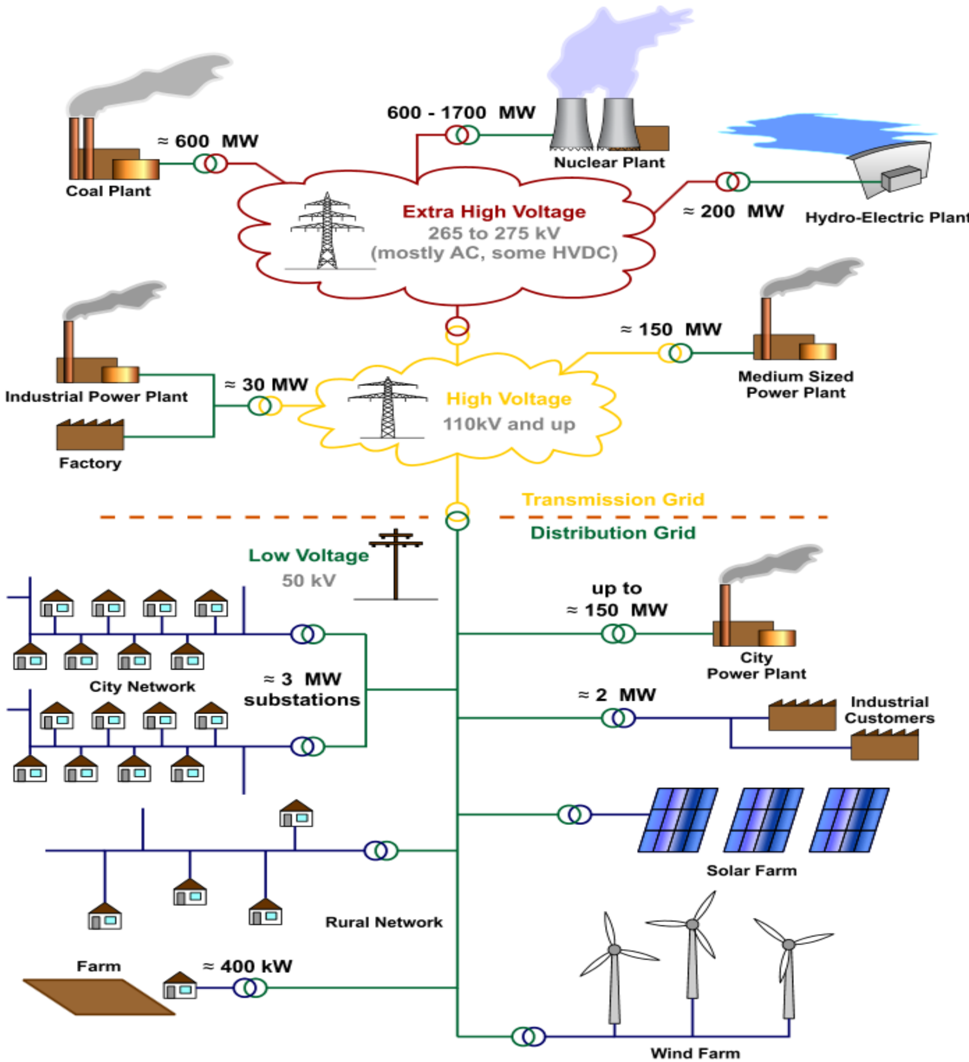
PA1 – Power Distribution Network Reconfiguration

Kaifeng Yang, Furong Ye

Universiteit Leiden
The Netherlands

Background -- (1)

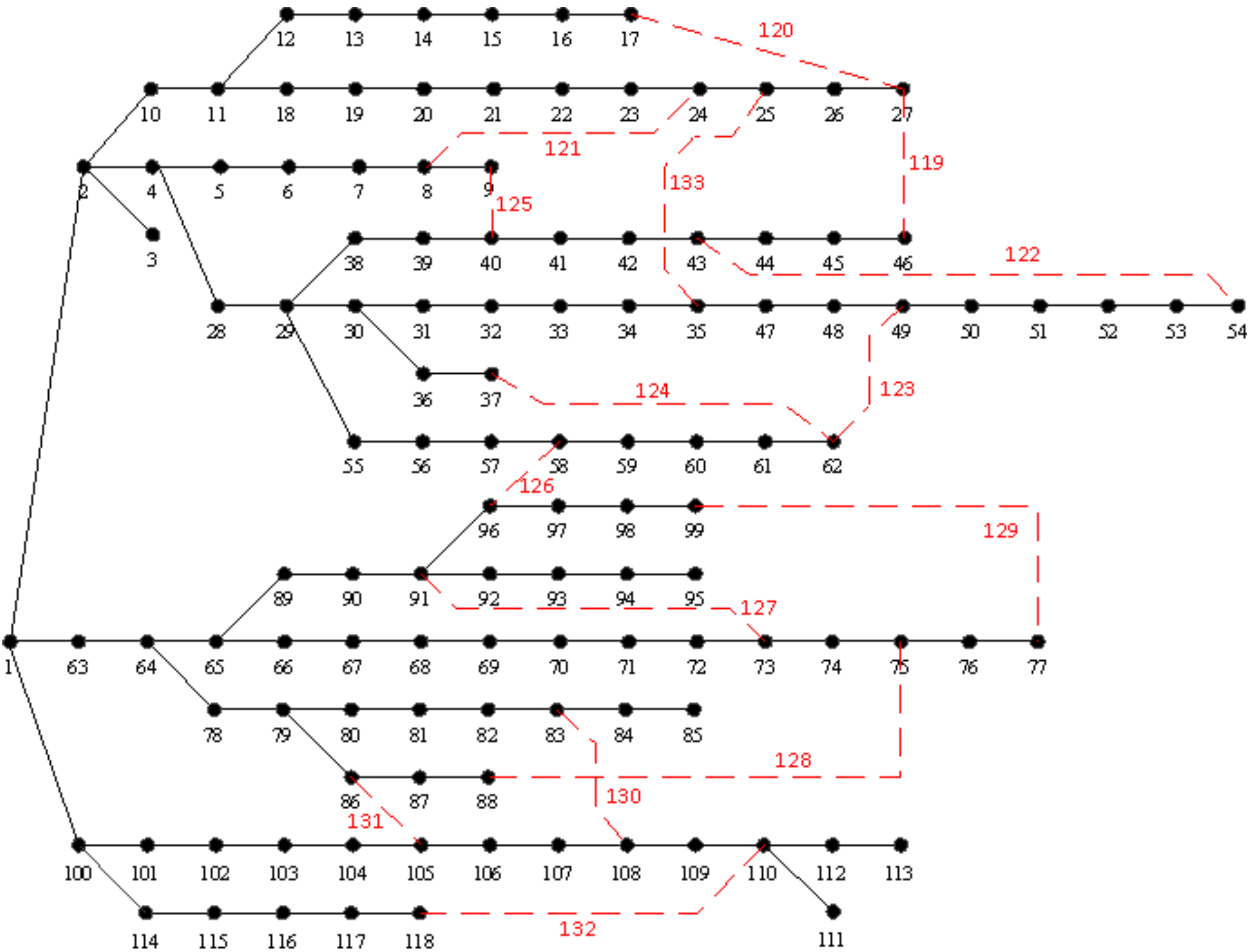
- Components of electrical grid, from plants to consumers:
  - generation
  - substation (increase V)
  - transmission line
  - substation (decrease V)
  - power distribution



Background -- (2)

- Why increase the voltage before the transmission?
  - To reduce active power loss
  - Power: $P = U \times I$
  - Powerloss: $P_{loss} = I^2 R$, where R is the resistance of the transmission line
  - R is determined by the material of the transmission line, length

Problem Definition – (1)



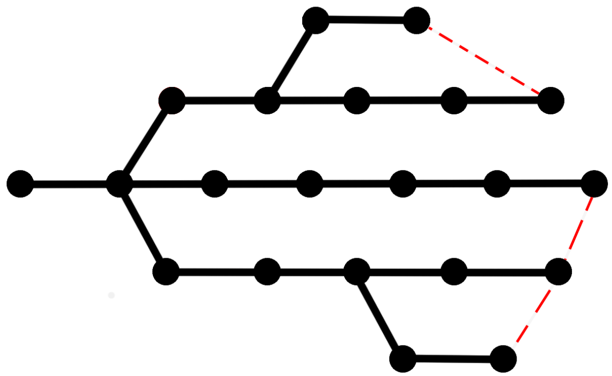
Initial configuration of the 119 test system

- Black solid lines: normally closed switches
- Red dashed lines: normally open switches. Why?
- Black nodes: transformers/ customers
- Node 1 is a substation/source
- Loop

Problem Definition – (2)

- **Definition**

Power network reconfiguration is the process of change the topology of the power network by operating the normally closed and normally open switches, for the purposes of minimization of the power loss and some other indicators.



- **Constraints**:
    1. Cycle free: avoid short circuit
    2. No separated component: make sure every customer can get power supply

Phenotype → genotype (1)

- How to represent/describe the topology of a power distribution network?

- Any ideas?

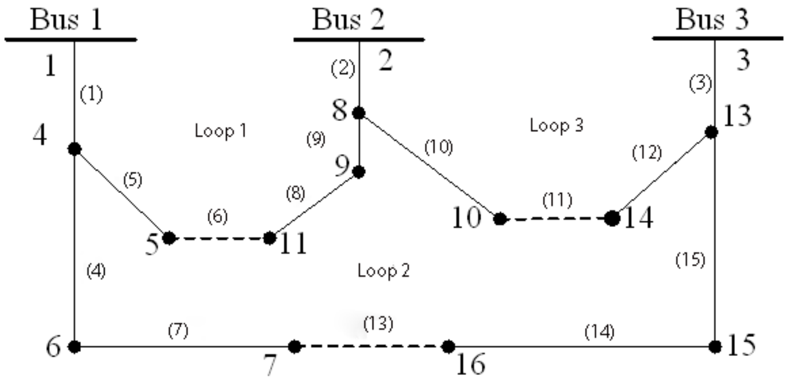
Phenotype → genotype (2)

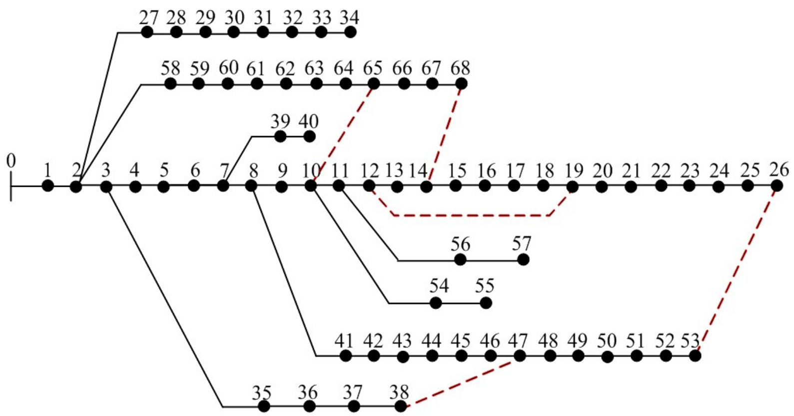
- Binary encoding
  - Use 0 and 1 to represent the status of each switch (open and close)
  - Easy and straightforward
  - Search space

    IEEE-16: 2^{12}=4096

    IEEE-69: 2^{73}

    119 system: 2^{132}
- Can we decrease the search space?





Phenotype → genotype (3)

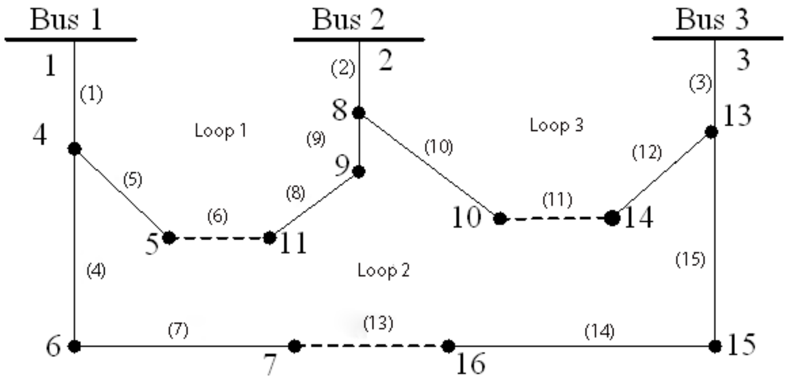
- Sequence encoding
  - Use an integer string to only represent the open
  - Search space

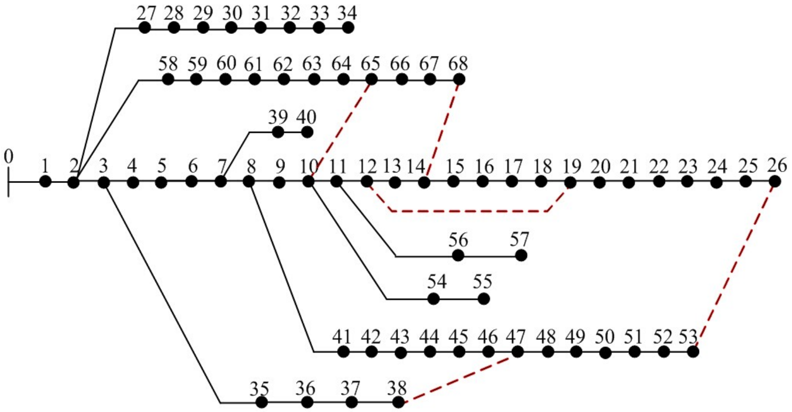
    IEEE-16: $4\times5\times3 = 60$

    IEEE-69: 1.79×10^{6}

    119 system: 1.44×10^{18}

- Increase the percentage of the feasible decision vectors in a search space





Feasible check – (1)

- Algorithm – deep check:

Step 1 Verify *N_line* equals *N_bus-1*;

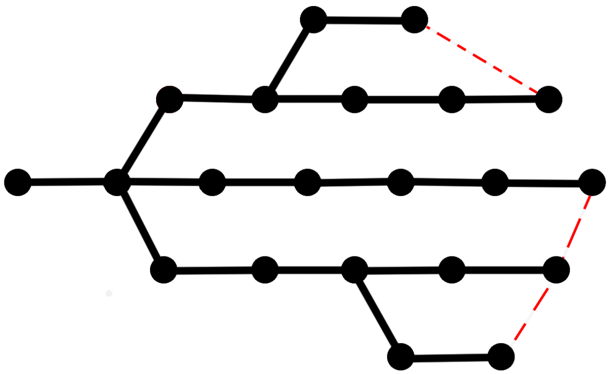
If not, not feasible;

else, go to step 2;

Step 2 Verify whether there exists a separated component;

if exists, not feasible;

else, feasible



Feasibility check

Feasible check – (2)

- Feasible check is done by **valid_119.m**:

  - Example: flag = valid_119(individual);

  - Input parameter a is a **column integer vector**;

  - flag = 1, if the “individual” is feasible

  - flag = 0, if the “individual” is infeasible

- Example
  - See matlab

Feasible check – (3)

- The input parameter "individual" is column integer vector, and it is formed like:

  [1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15].

- If you use binary coding system, you need to write your own converting function, which can convert a binary bitstring into a integer column vector, like:

[1 0 0 1 0 1 1 1 0 1 … 1 1 0 1 1 0 0 0 1]

Your own converting function, if needed

individual = [1; 2; 5; 2; 5; … ; 14; 20; 1; 15],

len(individual) = 15

Objective function – (1)

- (Active) Power loss:

$$\min f_{loss} = \sum_{i=1}^{b} k_i R_i \frac{P_i + Q_i}{V_i} = \sum_{i=1}^{b} k_i R_i \mid I_i \mid^2$$

subject to :

$$V_i^{min} \leq V_i \leq V_i^{max}$$

$$I_i \leq I_i^{max}, i = 1, \dots, b$$

If the i-th switch is closed, k_i is 1; otherwise, k_i is 0

Objective function – (2)

- Active power loss is calculated by **calculation_119.m**:

  - Example: fitness = calculation_119(a);

  - Input parameter a is a column integer vector;

- Example:
  - See matlab

Objective function – (3)

- The input parameter "individual" is column integer vector, and it is formed like:

  [1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15].

- If you use binary coding system, you need to write your own converting function, which can convert a binary bitstring into a integer column vector, like:

[1 0 0 1 0 1 1 1 0 1 … 1 1 0 1 1 0 0 0 1]

Your own converting function, if needed

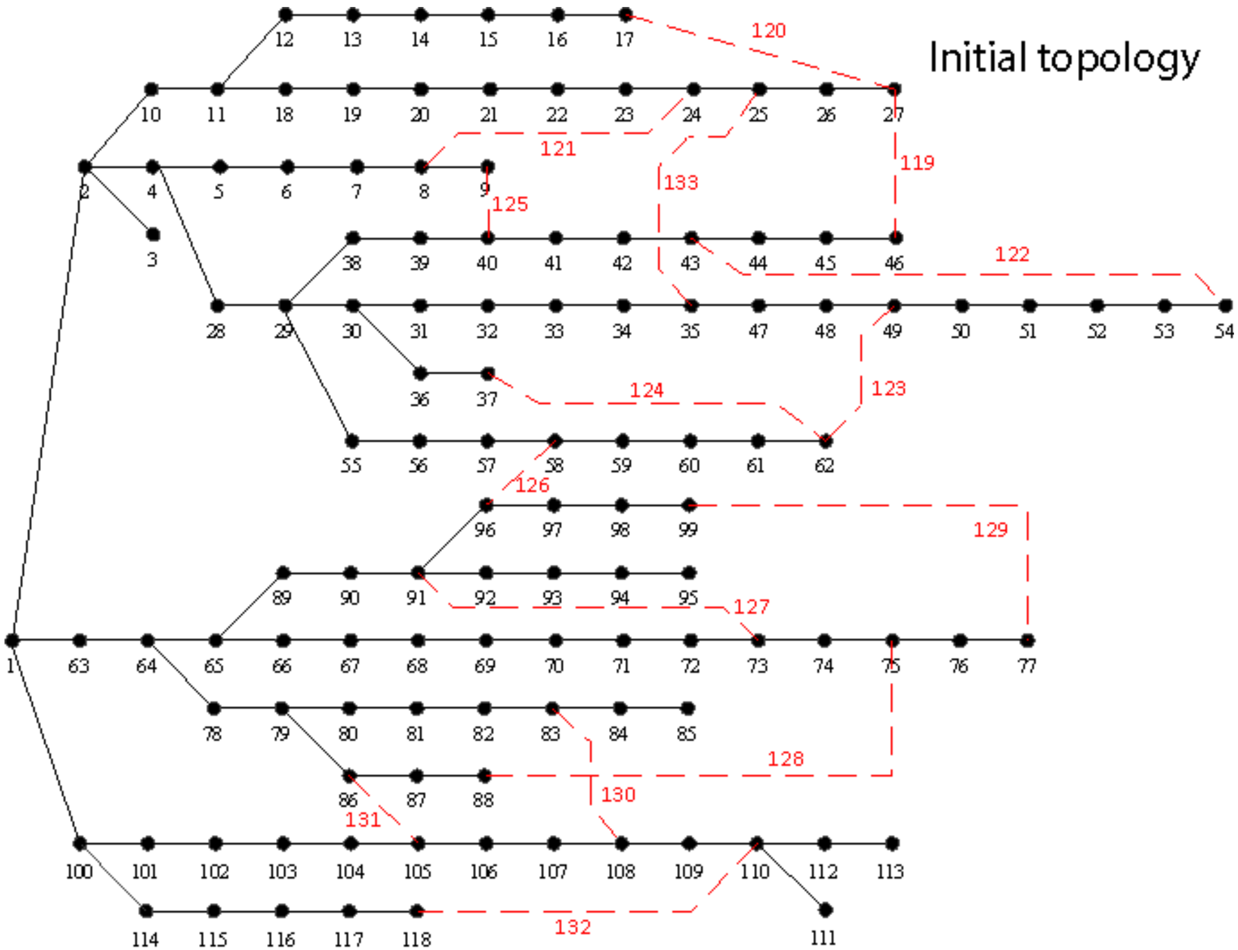
individual = [1; 2; 5; 2; 5; … ; 14; 20; 1; 15],

len(individual) = 15

Initial info.

- 119 test system:
  - 119 normally closed switches
  - 15 normally open switches
  - Power loads: 22709.7kW and 17041.1kVAr
  - Voltage: 11kV
  - Power loss: 1294.3kW

Results



Initial topology

The open switches in best result are: 42-43, 26-27, 23-24, 51-52, 62-49, 58-59, 39-40, 91-96, 71-72, 74-75, 97-98, 108-83, 105-86, 109-110 and 34-35.

Active power loss:

869.7271 kW

Provided – (1)

- **matpower4.1**
  - A MATLAB open source platform for power flow calculation;
  - Only use the package which provided by us;
  - Add the path of this folder to your MATLAB environment firstly;
  - The official website is: http://www.pserc.cornell.edu/matpower/
- **para119.mat :**
  - Basic parameters of 119 system:

    para_lb: [1; 1; … ; 1], the lower bound information for each loop, a decimal column vector

    para_ub: [ub1; ub2; … ; ub15], the upper bound information for each loop, a decimal column vector

    para_n: the number of loops in 119 system, an integer number 15
  - Load this file before your GA loop.

Provided – (2)

- Feasible check algorithm
  - flag = valid_119(individual)
  - Input **must** be a column integer vector

- Active power loss algorithm
  - fitness = calculation_119(individual);
  - Input **must** be a column integer vector

Basic framework of your source code

1. Initialize P(t);
2. Load “para119.mat”;
3. Add the path of “matpower4.1” into MATLAB environment;
4. Convert P(t) to Pd(t);
5. Feasible check for Pd(t) and Evaluate Pd(t);
6. **While not** terminate **do**
   1. P’(t) := select-mates(P(t));
   2. P’’(t) := crossover(P’(t), pm);
   3. P’’’(t) := mutate(P’’(t),pc);
   4. Pd(t) := convert(P’’’(t)); /* Convert a bitstring into a decimal column vector */
   5. **If** Pd(t) is feasible /* Feasible check */

      Evaluate(Pd(t)); /* Evaluation */

      Otherwise

      /* Write based on your own method */

      end
   6. P(t+1) := P’’’(t);
   7. t := t+1;

   Od
7. Output

Further readings:

- Electric power distribution
- Electric power transmission
- Power loss minimization in distribution system using network reconfiguration in the presence of distributed generation
- Power distribution network reconfiguration by evolutionary integer programming