Power Loss Reduction through Distribution Network Reconfiguration Using Feasibility-Preserving Simulated Annealing

19th International Scientific Conference on Electric Power Engineering

May 16-18, 2018, Brno, Czech Republic

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Slawomir Koziel
Alberto Landeros Rojas
Szczepan Moskwa

Introduction

- Distribution Network Reconfiguration (DNR) is a configurative process where the topology of the system is modified by switching normally open/closed switches with typical objective being power loss reduction, voltage quality improvement, reliability enhancement and/or load balancing.
- The optimization algorithm is sequential (not population-based) and exploits the techniques adopted from simulated annealing to avoid getting stuck in local optima.
- Proposed algorithm for distribution network reconfiguration for power loss reduction and voltage profile improvement implements the mechanisms for maintaining the radial network topology throughout the optimization process.

Introduction

» Comprehensive numerical studies were made using 33- and 69-bus distribution test system.

» The results obtained using FPSA have been compared with results reported in the literature, where other algorithms were used:

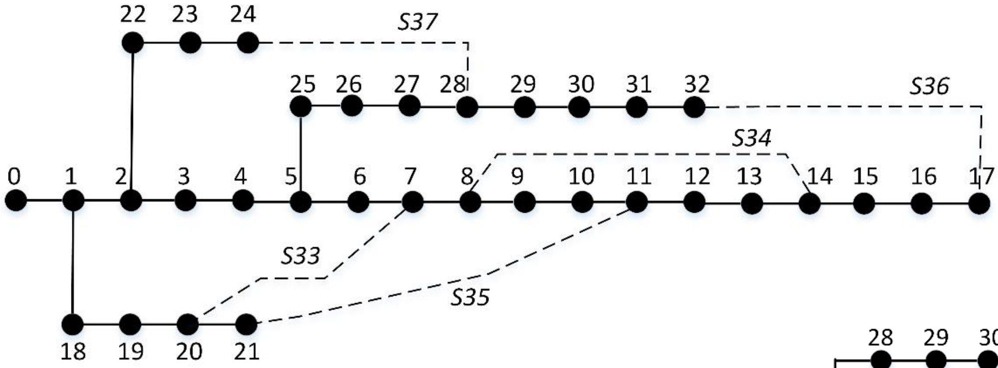
- CSA, FWA, HSA, RGA, ITS, GA for 33-bus system,
- CSA, GA, AC, IAICA for 69-bus system.

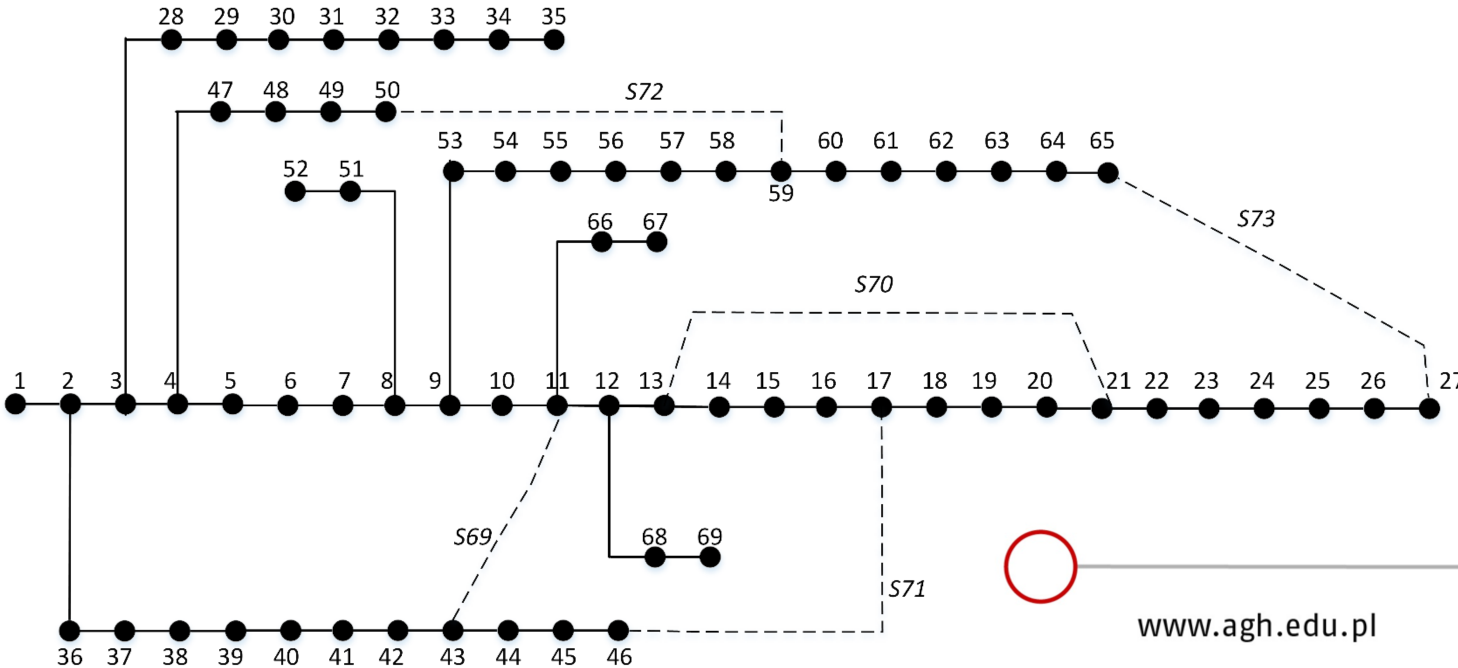
Acronyms:

CSA	– cuckoo search algorithm	ITS	– improved tabu search
FWA	– fireworks algorithm	GA	– genetic algorithm
HSA	– harmony search algorithm	AC	– ant colony algorithm
RGA	– refined genetic algorithm	IAICA	– improved adaptive imperialist competitive algorithm

33- and 69-bus distribution test system

Test Case	Initial Loss (kW)	Min V (p.u)	Initial open switches
33 Bus	200.745	0.9107	33, 34, 35,36, 37
69 Bus	223.725	0.9094	69, 70, 71, 72, 73





The L-N base voltage level for the 33 and 69 bus systems is set to be 12.66 kV. The power flow is solved by OpenDSS with a tolerance threshold of 0.005.

Power loss reduction by DNR

The reconfiguration process attempts to minimize the objective function:

$$F = \Delta P_{loss}^{rec} + \Delta V_D$$

The loss reduction index:

$$\Delta P_{loss}^{rec} = \frac{P_{loss}^{rec}}{P_{loss}^{init}}$$

P_{loss}^{rec} – the net power losses after reconfiguration

P_{loss}^{init} – the initial net power losses

The indicator of the deviation in voltage levels after reconfiguration:

$$\Delta V_D = \max_{i=1,..N_{bus}} \left(\frac{V_1 - V_i}{V_1} \right)$$

V_1 – the voltage level of the source bus
V_i – the bus voltage level

Power loss reduction by DNR

The net power losses of the distribution system:

$$P_{loss} = \sum_{i=1}^{Nbr} R_i \times \left(\frac{P_i^2 + Q_i^2}{\left| V_i^2 \right|} \right)$$

P_i – the active power flowing out of the bus *i*
Q_i – the reactive power flowing out of the bus *i*
R_i – the resistance of line segment *i*
V_i – the voltage at the *i*th bus
N_{br} – the number of power line segments

Power loss reduction by DNR

Power flow analysis of the test distribution systems is made by the open distribution system simulator – OpenDSS.

The assumption to perform power flow analysis:

- all line segments in the system have a sectionalizing switch,
- the number of switch controls N_s is a sum of the number of sectionalizing switches N_{sec} and tie switches N_{ts},
- the switch status is represented in a binary string $\boldsymbol{x}$ of size N_s (0 - an open switch; 1 - a closed switch).

Optimization methodology

The proposed feasibility-preserving simulated annealing (FPSA) algorithm is loosely based on simulated annealing paradigm. It features the mechanisms for introducing local network reconfigurations as well as random changes accepted under certain probability.

Notation used in algorithm:

$\boldsymbol{x}^{(0)}$	– the initial solution,
T and T_0	– the current and the initial system temperatur,
$r \in (0,1]$	– a random number drawn with uniform probability distribution,
i_{max}	– the maximum number of algorithm iterations.

Optimization methodology - Algorithm Structure

1. The function *random_radial_configuration*() produces a random network configuration, which is then compared to the current solution $\boldsymbol{x}^{(i)}$ and accepted under two conditions: (i) it exhibits a better value of the objective function than $F(\boldsymbol{x}^{(i)})$, or (ii) it is worse but the condition $\exp((F_{current} - F(\boldsymbol{x}^{rand}))/T) > r$ is satisfied for the random number r.

1. Set $i = 0$ (iteration counter);
2. Set $F_{best} = F_{current} = F(\boldsymbol{x}^{(0)})$; $\boldsymbol{x}^{best} = \boldsymbol{x}^{(0)}$;
3. $\boldsymbol{x}^{rand}$ = *random_radial_configuration*();
4. **If** $F(\boldsymbol{x}^{rand}) < F_{current}$ OR $\exp((F_{current} - F(\boldsymbol{x}^{rand}))/T) > r$ **then** set $\boldsymbol{x}^{(i+1)} = \boldsymbol{x}^{rand}$ and $F_{current} = F(\boldsymbol{x}_{rand})$ and go to 8;
5. $\boldsymbol{x}^{local} = local\_search(\boldsymbol{x}^{(i)})$;
6. **If** $F(\boldsymbol{x}^{local}) < F_{current}$ **then** set $\boldsymbol{x}^{(i+1)} = \boldsymbol{x}^{local}$ and $F_{current} = F(\boldsymbol{x}^{local})$ and go to 8;
7. $\boldsymbol{x}^{(i+1)} = random\_change(\boldsymbol{x}^{(i)})$; $F_{current} = F(\boldsymbol{x}^{(i+1)})$;
8. **If** $F_{current} < F_{best}$ **then** set $\boldsymbol{x}^{best} = \boldsymbol{x}^{(i+1)}$; $F_{best} = F_{current}$;
9. $T = T_0(1 - i/i_{max})$;
10. Set $i = i + 1$;
11. **If** $i < i_{max}$ **then** go to 3, **else** END (return $\boldsymbol{x}^{best}$);

Optimization methodology - Algorithm Structure

2. The main search step is the local search which aims at finding a (local) modification of the current configuration $\boldsymbol{x}^{(i)}$ that results in maximum reduction of the objective function value. In case the local search is successful, the resulting network configuration is accepted, otherwise, a (local) random change is introduced (function *random_change*()).

1. Set $i = 0$ (iteration counter);
2. Set $F_{best} = F_{current} = F(\boldsymbol{x}^{(0)})$; $\boldsymbol{x}^{best} = \boldsymbol{x}^{(0)}$;
3. $\boldsymbol{x}^{rand} = \textit{random\_radial\_configuration}()$;
4. **If** $F(\boldsymbol{x}^{rand}) < F_{current}$ OR $\exp((F_{current} - F(\boldsymbol{x}^{rand}))/T) > r$ **then** set $\boldsymbol{x}^{(i+1)} = \boldsymbol{x}^{rand}$ and $F_{current} = F(\boldsymbol{x}_{rand})$ and go to 8;
5. $\boldsymbol{x}^{local} = \textit{local\_search}(\boldsymbol{x}^{(i)})$;
6. **If** $F(\boldsymbol{x}^{local}) < F_{current}$ **then** set $\boldsymbol{x}^{(i+1)} = \boldsymbol{x}^{local}$ and $F_{current} = F(\boldsymbol{x}^{local})$ and go to 8;
7. $\boldsymbol{x}^{(i+1)} = \textit{random\_change}(\boldsymbol{x}^{(i)})$; $F_{current} = F(\boldsymbol{x}^{(i+1)})$;
8. **If** $F_{current} < F_{best}$ **then** set $\boldsymbol{x}^{best} = \boldsymbol{x}^{(i+1)}$; $F_{best} = F_{current}$;
9. $T = T_0(1 - i/i_{max})$;
10. Set $i = i + 1$;
11. **If** $i < i_{max}$ **then** go to 3, **else** END (return $\boldsymbol{x}^{best}$);

Optimization methodology - Algorithm Structure

3. The best solution is updated in Step 8.

1. Set $i = 0$ (iteration counter);
2. Set $F_{best} = F_{current} = F(\boldsymbol{x}^{(0)})$; $\boldsymbol{x}^{best} = \boldsymbol{x}^{(0)}$;
3. $\boldsymbol{x}^{rand} = random\_radial\_configuration()$;
4. **If** $F(\boldsymbol{x}^{rand}) < F_{current}$ OR $\exp((F_{current} - F(\boldsymbol{x}^{rand}))/T) > r$ **then** set $\boldsymbol{x}^{(i+1)} = \boldsymbol{x}^{rand}$ and $F_{current} = F(\boldsymbol{x}_{rand})$ and go to 8;
5. $\boldsymbol{x}^{local} = local\_search(\boldsymbol{x}^{(i)})$;
6. **If** $F(\boldsymbol{x}^{local}) < F_{current}$ **then** set $\boldsymbol{x}^{(i+1)} = \boldsymbol{x}^{local}$ and $F_{current} = F(\boldsymbol{x}^{local})$ and go to 8;
7. $\boldsymbol{x}^{(i+1)} = random\_change(\boldsymbol{x}^{(i)})$; $F_{current} = F(\boldsymbol{x}^{(i+1)})$;
8. **If** $F_{current} < F_{best}$ **then** set $\boldsymbol{x}^{best} = \boldsymbol{x}^{(i+1)}$; $F_{best} = F_{current}$;
9. $T = T_0(1 - i/i_{\max})$;
10. Set $i = i + 1$;
11. **If** $i < i_{\max}$ **then** go to 3, **else** END (return $\boldsymbol{x}^{best}$);

Optimization methodology - Function *random radial configuration*

- Random radial configuration is generated by starting from an initial configuration corresponding to all switches being closed and sequentially opening randomly selected switches until radial configuration is obtained.
- After selecting and opening a switch, the network graph connectivity is checked and in case the graph is not connected, the switch is closed, and replaced by another (randomly selected) switch.
- This is sufficient because the configuration obtained after opening N_{ts} switches is radial if and only if the corresponding graph is connected.

Optimization methodology - Function *local search*

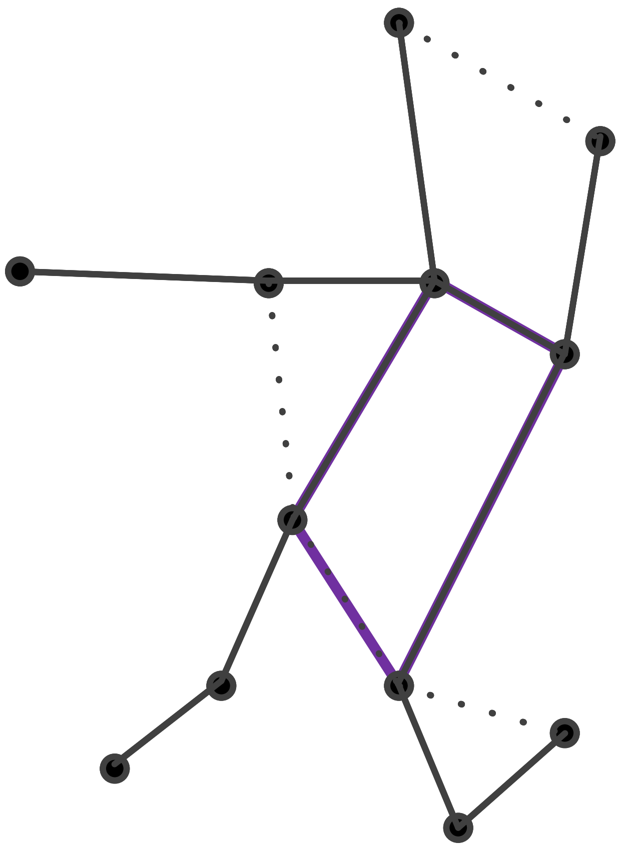
» The local search step sequentially modifies the network configuration by closing all open switches, one by one, and opening adjacent connections, looking for modifications that are the most beneficial from the point of view of objective function reduction.

» The local search preserves radial configuration of the network.

» Its computational cost is $2N_{ts}$ evaluations of the objective function.

1. For all open switches s_k, $k = 1, \ldots, N_{ts}$:
   - Close the switch s_k;
   - Identify the loop created by closing the switch;
   - Find two connections on the loop, adjacent to s_k, and the corresponding switches $s_{k.j}$, $j = 1,2$;
   - For $j = 1, 2$, open the switch $s_{k.j}$ and calculate objective function $F_{k.j}$ of the resulting radial network $\boldsymbol{x}^{(k.j)}$;
1. **If** $\min\{k = 1, \ldots, N_{ts}; j = 1, 2 : F_{k.j}\} < F_0$ **then** return $\boldsymbol{x}^{(k.j)}$ that realizes the minimum; **else** return $\boldsymbol{x}^{(i)}$;

Optimization methodology
- Function *random change*



1. Part of the radial network with one of the switches randomly selected to be closed and create connection.

2. A loop created by closing the switch.

3. One of the connections along the loop randomly selected to be opened, thus retaining the radial configuration of the network.

Results and Benchmarking

Statistics of the optimization results:
33-bus system

Method	N_{evals}	Best (kW)	Average (kW)	Worst(kW)	STD	Fitness Value
FPSA	500	140.3350	140.3350	140.3350	0.0000	0.7607
CSA	3000	139.8476	N/A*	N/A*	N/A*	0.7618
FWA	1000	140.3350	147.02	157.243	N/A*	0.7607
HSA	2500	142.8780	153.82	197.01	N/A*	0.7810
RGA	N/A*	139.8476	166.51	200.34	N/A*	0.7618
ITS	600	142.8780	165.1	198.22	N/A*	0.7810
GA	21000	139.8476	167.82	204.68	N/A*	0.7618

* Relevant data has not been provided

Results and Benchmarking

Statistics of the optimization results: 69-bus system

Method	N_{evals}	Best (kW)	Average (kW)	Worst(kW)	STD	Fitness Value
FPSA	2000	98.9299	98.9798	99.9276	0.0013	0.4992
CSA	3000	98.9418	N/A*	N/A*	N/A*	0.4993
GA	900	98.9418	101.34	104.73	N/A*	0.4993
AC	900	99.1225	103.18	110.28	N/A*	0.5001
IAICA	900	98.9418	100.57	104.25	N/A*	0.4993

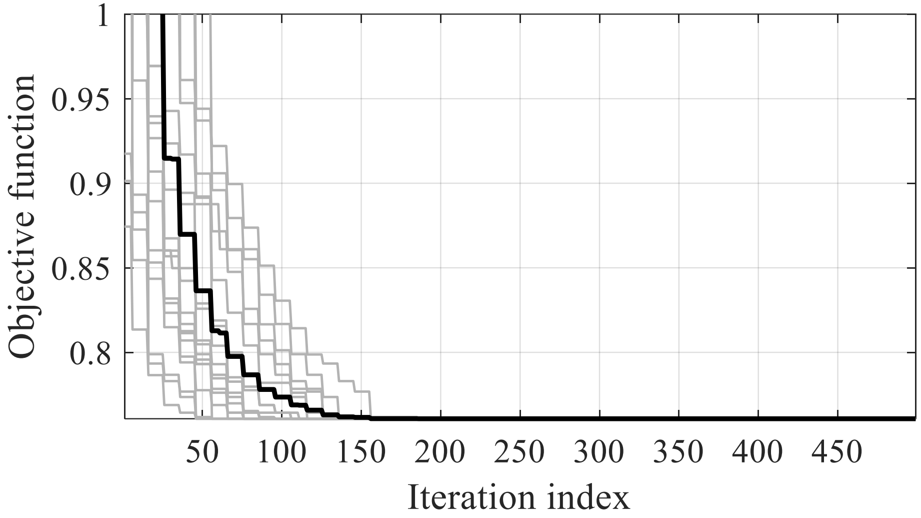
* Relevant data has not been provided

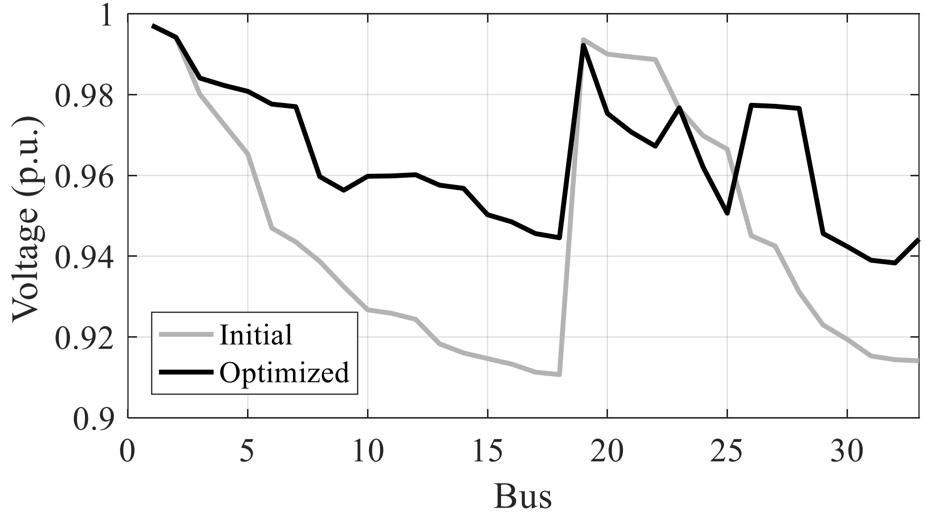
Results and Benchmarking

Optimization history for considered test cases

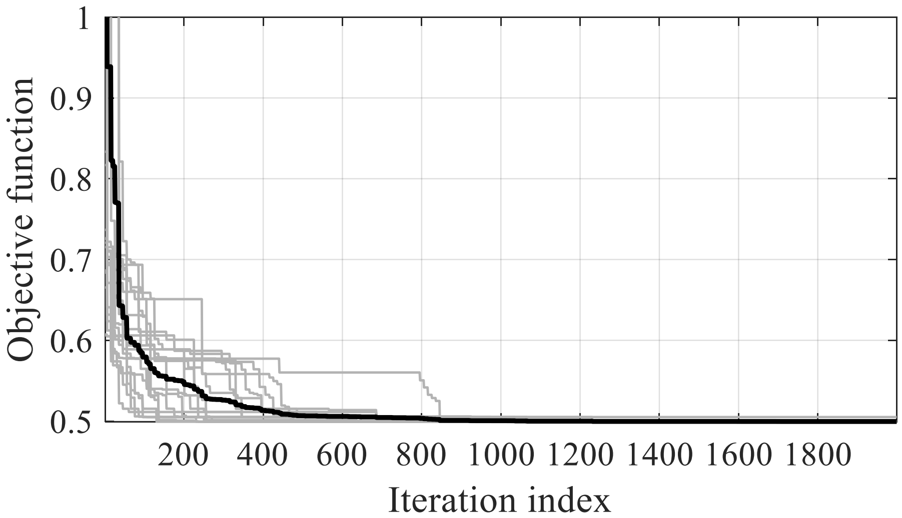
Network voltage profile before and after reconfiguration

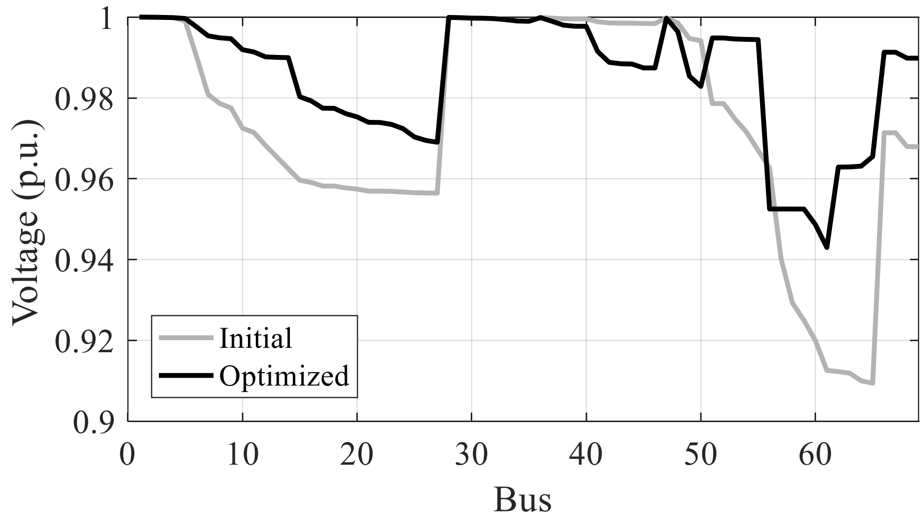
33-bus system





69-bus system





Conclusion

» Our approach is a sequential stochastic that adopts certain mechanisms from simulated annealing but also customized procedures to perform local improvement of the cost function by single-connection network reconfigurations as well as preservation of solution feasibility (radial configuration).

» Extensive numerical experiments and benchmarking carried out for the standard 33- and 69-bus systems indicate the superiority of the proposed approach over comparative methods.

» The advantages of our methodology include robustness of the optimization process (in terms of improved repeatability of results) but also lower computational cost.

AGH

Thank you for your attention