Politecnico di Torino
Dipartimento di Ingegneria Elettrica

DISTRIBUTION SYSTEM OPTIMISATION

Prof. Gianfranco Chicco

Lecture at the Technical University "Gh. Asachi", Iaşi, Romania

26 October 2010

Outline

- *Classification* of the distribution system optimisation problems
- *Objective* functions and *constraints*
- Types of optimisation *algorithms*
  - √ *deterministic*
  - √ *meta-heuristic*
- Minimum loss *reconfiguration*
  - √ *iterative improvement*
  - √ *simulated annealing*

Classification of the optimisation problems

- Optimal *reconfiguration* in *normal* operating conditions
  - √ choice of the *branches to maintain open* in order to supply all the loads from a radial network
- Optimal *reconfiguration* in *emergency* conditions
  - √ identification of the best *strategy* to supply the loads after a fault in order to limit the effects of the outage
- Optimal *operational planning* (at constant load)
  - √ choice of the most effective set of *structural planning actions* among a set of proposed planning actions
- Optimal *expansion planning* (at variable load)
  - √ selection of the most effective actions for *expanding* the distribution system according to the foreseeable spatial evolution of the load, with strategic, economic and social implications

Optimisation sub-problems

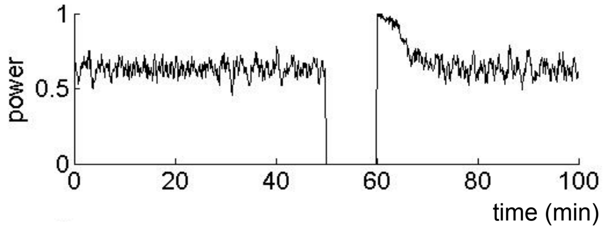
- Optimal location of the power factor compensation devices
  - √ choice of the number, location and size of the devices for power factor correction and voltage profile improvement
- Optimal choice of the closed terminal of the open branches
  - √ for a distribution system operated with isolated neutral, the open branches are kept *closed* at the terminal with the *highest automation level* in order to close the branch more rapidly in case of need
  - √ when the automation level of the nodes at the two terminals of the branch is *the same*, the choice of the terminal to maintain open can be performed in order to *optimise* a suitable objective function
- Optimal strategy for service restoration
  - √ in the presence of particular types of loads, such as *thermostatically-controlled* loads (e.g., for space heating or conditioning), the effectiveness of the service restoration depends on establishing a feasible *sequence* of re-connection of the feeders, in order to avoid the effects of the *Cold Load Pickup*

Cold Load Pickup

- The effects *Cold Load Pickup* occur when a group of loads (e.g., heating loads) controlled by thermostats is subject to strong voltage reductions or interruptions of relatively high duration
- Normally, the loads under thermostatic control operate in *intermittent* mode, with average power lower than the maximum due to the *duty-cycle*
- An *aggregation* of these loads exhibits a total power always lower than the maximum power due to the *time diversity* (non synchronised operation among the thermostats)
- After a *supply interruption*, the power falls to zero and the temperature of the controlled devices falls down with respect to the temperature range of thermostat control
- When the supply is restored, the (heating) loads start from "cold" conditions and are switched on *all together*

Cold Load Pickup

- The simultaneous switch-on causes a strong *increase* in the instantaneous total load power, with respect to the normal power levels
- Such power increase may have a *relatively long duration* before its reduction to normal values, thus causing *temporary overloads* in the system and possible trip of protections
- The effects of the Cold Load Pickup have to be taken into account during the system *restoration* in case of high contribution of loads controlled by thermostats to the total load
- The switch-on instants have to be scheduled at *successive periods*, to avoid excessive overloads



Objectives of the optimisation

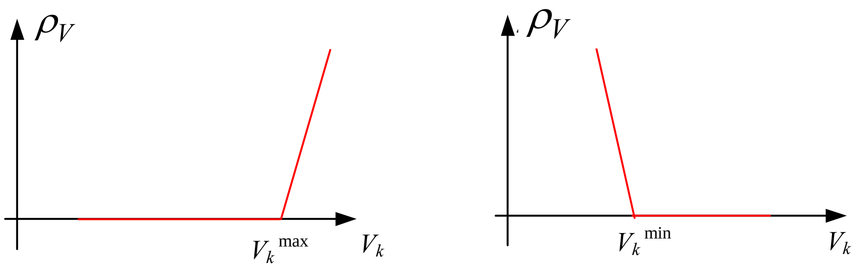
- The objectives can be of technical and/or economic nature, e.g.:
  1. *loss* minimisation
  2. *cost* minimisation (concerning investment, operation and maintenance costs);
  3. optimisation of suitable *reliability indicators* (e.g., reduction of the expected number of interruptions, duration of the interruptions or energy not supplied in a given time period);
  4. *constraint satisfaction* (e.g., in the service restoration phase during emergencies)
- *Multi-objective* formulations are possible, by building a *composite* objective function as the *weighted sum* of the single objectives, assigning suitable weights to each objective

Constraints

- Distribution systems *operation* is constrained by:
  - *max voltage* magnitude at each load node
  - *min voltage* magnitude at each load node
  - *max phase-earth* fault current at each node
  - *max 3-phase short-circuit* current at each node
  - *thermal* current *rating* limit for each branch
  - avoidance of *cascade* circuit breakers (due to lack of selectivity)
  - maximum number of *switching operations* from the initial to the final configuration (to build a feasible trajectory of operations that make it possible the transition between the two radial configurations)
  - ensuring *independent supplies* to the customers that set up a contract with the supplier for availability of a *second delivery point*

Bilateral constraints

- Each bilateral constraint is transformed into a *pair* of unilateral constraints
- Example for *voltage* magnitude at load node *k*



Penalised objective function

- The effects of constraint violation can be added to the objective function to form a *penalised objective function*
- The use of penalised objective functions allows for *accepting temporary violations* of the constraints during the solution process
- Each constraint violation is associated to a constraint *multiplier* higher than zero (the multiplier is zero if the constraint is not violated)
- The multipliers are *user-defined*
- If the multipliers are *large* enough, no violation should occur in the final solution
- If one or more multipliers are relatively *small*, residual violations could occur in the final solution

Penalised objective function

- Example of penalised objective function for *minimum loss* reconfiguration

$$f(\mathbf{x}) = \sum_{b \in \boldsymbol{B}} R_b I_b^2 \left(1 + \sum_{\upsilon \in \boldsymbol{V}} \rho_\upsilon \left(X_\upsilon^{\lim} - X_\upsilon \right)^2 \right)$$

where:

x – system configuration

B – set of branches

V – set of constrained variables

R_b – branch resistance

I_b – branch current magnitude

X_υ – generic constrained variable

ρ_υ – constraint multiplier

Types of optimisation algorithms

- The optimisation problems listed above, applied to a *large* distribution systems, may be characterised by
  - √ objective functions with *non-smooth* surfaces corresponding to the solution points, and several *local minima*
  - √ *combinatorial explosion* of the number of cases to be evaluated to search for the global optimum
- The algorithms used should be able to perform efficient *global optimisation* on a combinatorial problem
- *Two types* of methods are used
  - √ *deterministic methods*, in which the solution strategy is driven by well-identified rules, with *no random* components
  - √ *meta-heuristic methods*, whose evolution depends on *random* choices carried out during the evolution of the solution process; in several cases these methods aim at represent some *natural* behaviour

Deterministic methods

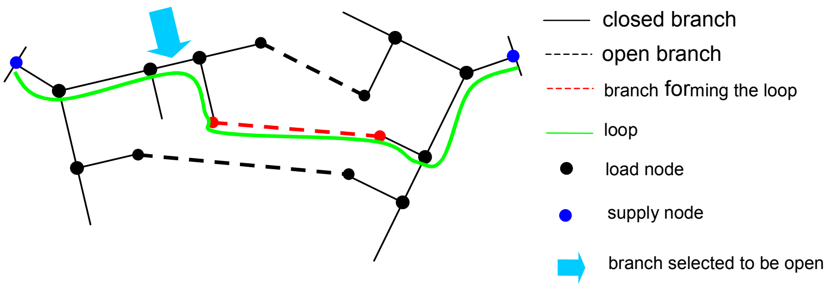
- *Exhaustive search*
  - √ used when the *number* of solutions is *known*, and it possible to compute the objective function for all the cases with *reasonable computation time*
- *Branch and bound* techniques
  - √ used when the number of solutions is *known*, and it possible to *limit* the number of solutions computed by *eliminating* a number of cases that necessarily provide worse solutions
- *Iterative improvement*
  - √ search strategy that proceeds along a *trajectory* of the solutions by *successive* modifications, until the situation in which the procedure cannot provide further modifications
- *Tabu search*
  - √ search strategy that allows for *marking* some solutions as "*tabu moves*" in order to avoid to reach them during a number of successive steps of the solution process

Meta-heuristic methods

- *Genetic algorithms* and *evolutionary computation*
  - √ apply the principles of *genetic evolution*, with reproduction by selection of the fittest elements, crossover and mutation
- *Simulated annealing*
  - √ simulation of the *annealing process* in which a melting metal is slowly cooled to solidify in its *minimum energy* state
- *Ant Colony Search*
  - √ based on the behaviour of groups of *ants* while searching for food
- *Particle Swarm Optimisation*
  - √ reproduction of the *social behaviour* and organized movements change without collisions, as it happens in the choreography of the bird flocks
- *Immune* and *clonal selection*-based algorithms
  - √ applications of some principles of *immunology*

Branch-exchange mechanism

- Essential mechanism applied starting from an *initial radial structure*, in order to reach only *radial* configurations, requiring the following steps:
  - √ *selection* of a branch belonging to the list of the *open* branches
  - √ the selected branch is *closed*, forming a *loop*
  - √ *selection* of the branch to be *open* among the branches of the loop



Radial distribution network configurations

- The *number* of branches to maintain *open* in a radial network with *redundant* connections is *fixed*

$$\Lambda = B - N + S$$

where

Λ = number of redundant branches

B = number of branches

N = number of nodes

S = number of supply points

Radial distribution network configurations

- However, it is not sufficient to open Λ branches chosen *arbitrarily* to form a radial network, as *loops* or *islands* with nodes not connected could remain in the network
- The formation of radial configurations is subject to the constraint of reaching all the nodes (to supply every load)

Kirchhoff's Theorem

- The calculation of the *number* of possible radial configurations extracted from a meshed structure has been first done by Kirchhoff
- The Kirchhoff's theorem can be applied to determine the number of *radial* configurations extracted from a *weakly meshed* structure of an electrical distribution system
- It is necessary to calculate the *determinant* of a matrix **Q*** obtained from the *Laplacian matrix* of the network by deleting one row and one column

Kirchhoff’s Theorem

- Laplacian Matrix

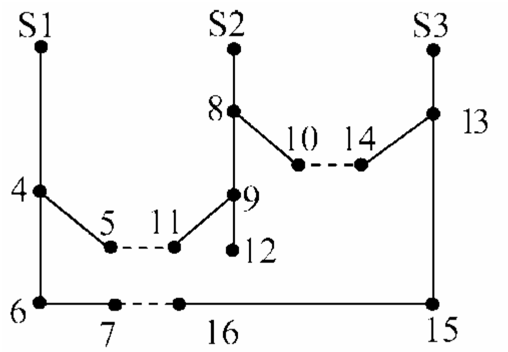
The *Laplacian matrix* is a square matrix $\mathbf{L} := (l_{i,j})_{nxn}$ defined as:

$$l_{i,j} := \begin{cases} n.\text{ of branches connected to node i} & \text{for } i = j \\ -1 & \text{if } i \neq j \text{ and node } i \text{ adjacent to node } j \\ 0 & \text{otherwise} \end{cases}$$

NOTE: The sum of elements of each row and each column is zero

Kirchhoff’s Theorem

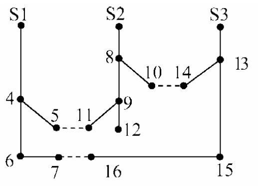
- Test System A [1]



[1] S. Civanlar, J. Grainger, H. Yin, and S. Lee, “Distribution feeder reconfiguration for loss reduction,” *IEEE Trans. Power Del.*, vol. 3, no. 3, pp. 1217–1223, Aug. 1988.

Kirchhoff's Theorem

- Laplacian matrix for Test System A



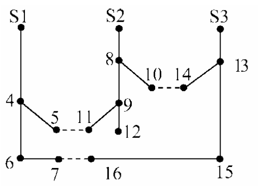
IOTE: S1, S2 and S3 are included in a single node 0

NOTE: node 12 is connected to node 9 througt a single branch: it is possible to eliminate node 12 and the branch connecting node 12 to node 9

	0	4	5	6	7	8	9	10	11	13	14	15	16
0	3	-1	0	0	0	-1	0	0	0	-1	0	0	0
4	-1	3	-1	-1	0	0	0	0	0	0	0	0	0
5	0	-1	2	0	0	0	0	0	-1	0	0	0	0
6	0	-1	0	2	-1	0	0	0	0	0	0	0	0
7	0	0	0	-1	2	0	0	0	0	0	0	0	-1
8	-1	0	0	0	0	3	-1	-1	0	0	0	0	0
9	0	0	0	0	0	-1	2	0	-1	0	0	0	0
10	0	0	0	0	0	-1	0	2	0	0	-1	0	0
11	0	0	-1	0	0	0	-1	0	2	0	0	0	0
13	-1	0	0	0	0	0	0	0	0	3	-1	-1	0
14	0	0	0	0	0	0	0	-1	0	-1	2	0	0
15	0	0	0	0	0	0	0	0	0	-1	0	2	-1
16	0	0	0	0	-1	0	0	0	0	0	0	-1	2

Kirchhoff's Theorem

- Matrix **Q*** for Test System A



Q* =

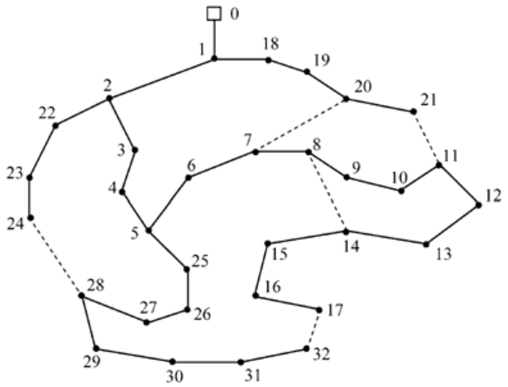
	0	4	5	6	7	8	9	10	11	13	14	15	16
0	3	-1	0	0	0	-1	0	0	0	-1	0	0	0
4	-1	3	-1	-1	0	0	0	0	0	0	0	0	0
5	0	-1	2	0	0	0	0	0	-1	0	0	0	0
6	0	-1	0	2	-1	0	0	0	0	0	0	0	0
7	0	0	0	-1	2	0	0	0	0	0	0	0	-1
8	-1	0	0	0	0	3	-1	-1	0	0	0	0	0
9	0	0	0	0	0	-1	2	0	-1	0	0	0	0
10	0	0	0	0	0	-1	0	2	0	0	-1	0	0
11	0	0	-1	0	0	0	-1	0	2	0	0	0	0
13	-1	0	0	0	0	0	0	0	0	3	-1	-1	0
14	0	0	0	0	0	0	0	-1	0	-1	2	0	0
15	0	0	0	0	0	0	0	0	0	-1	0	2	-1
16	0	0	0	0	-1	0	0	0	0	0	0	-1	2

det(**Q***) = 190

190 radial configurations of the network are possible

Kirchhoff's Theorem

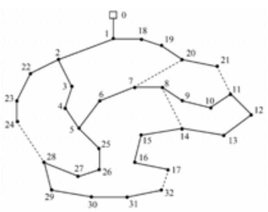
- Test System B [2]



[2] M. E. Baran and F. F.Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Trans. Power Del.*,vol. 4, no. 4, pp. 1401–1407, Nov. 1989.

Kirchhoff's Theorem

- Laplacian matrix for the Test System B Network



NOTE:

- node 0 is connected to node 1 through a single branch
- It is possible to eliminate node 0 and the branch connecting node 0 to node 1 (node 1 becomes the supply node)

Kirchhoff's Theorem

- Laplacian matrix for Test System B

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0
2	-1	3	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0
3	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	-1	3	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0
6	0	0	0	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	-1	3	-1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	-1	3	-1	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	-1	3	-1	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	-1	0	0	0	0	-1	3	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2	0	0	0	0	0	0	0	0	0	0	0	0	0
18	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	-1	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	-1	3	-1	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	-1	2	0	0	0	0	0	0	0	0	0
22	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	-1	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2	-1	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2	0	0	0	-1	0	0
25	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	-1	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2	-1	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2	-1	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	-1	3	-1	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2	-1
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2

Kirchhoff's Theorem

- Matrix **Q*** for the Test System B Network

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0
2	-1	3	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0
3	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	-1	3	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0
6	0	0	0	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	-1	3	-1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	-1	3	-1	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	-1	3	-1	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	-1	0	0	0	0	-1	3	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2	0	0	0	0	0	0	0	0	0	0	0	0	0
18	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	-1	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2	-1	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	-1	3	-1	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	-1	2	0	0	0	0	0	0	0	0	0
22	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	-1	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2	-1	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2	0	0	0	-1	0	0
25	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	-1	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2	-1	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2	-1	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	-1	3	-1	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2	-1
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	2

Kirchhoff's Theorem

- Matrix **Q*** for Test System B

$$\det(\mathbf{Q}^*) = 50751$$

⬇

50751 radial configurations of the network are possible

Minimum loss reconfiguration

- ***Formulation of the optimisation problem***
  - identification of the state of the *switches* at the branch terminals
  - constrained *combinatorial* problem
  - deterministic methods *reject* any solution with violations
  - meta-heuristic methods use of a *penalised* objective function

- ***Definition of the sets***
  - ***B*** (branches)
  - ***K*** (nodes)
  - ***S*** (supply substations)
  - ***X*** (radial configurations)
  - ***W*** (switches)

Constraints

$X \in \boldsymbol{X}$		radial network
$V_k - V_k^{\max} \le 0$	$k \in \boldsymbol{K}$	min voltage magnitude
$V_k^{\min} - V_k \le 0$	$k \in \boldsymbol{K}$	max voltage magnitude
$I_k^{ef} - I_k^{ef,\max} \le 0$	$k \in \boldsymbol{K}$	max phase-earth fault current
$I_k^{sc} - I_k^{sc,\max} \le 0$	$k \in \boldsymbol{K}$	max three-phase short circuit current
$I_b^{th} - I_b^{th,\max} \le 0$	$b \in \boldsymbol{B}$	branch current thermal rating
$n_{ck}(path(k,s)) - 1 = 0$	$k \in \boldsymbol{K}$	avoidance of cascade circuit breakers
$n - n^{\max} \le 0$		max number of switching operations
$n_{ds} = 0$		independence of double supplies

Iterative improvement algorithm

- A *radial* and *feasible* initial configuration (no constraint violation) is required
- The algorithm consists of an *iterative process*
- At each *iteration*
  - √ the *list* of the open branches is formed
  - √ the open branches are closed *one at a time*
  - √ after each closure, a radial configuration is formed by applying the *branch-exchange* mechanism: each branch belonging to the loop is open (one at a time) computing the corresponding objective function (exploring all the possibilities)
  - √ if there is an *improvement* of the objective function, the corresponding configuration becomes the best configuration
  - √ the process *stops* if at the end of the iteration no improvement of the objective function has been found

Iterative improvement algorithm

- The proposed method analyses a *large set* of configurations
- Each configuration includes the computation of the *objective function* (power flow and loss calculation)
- The calculation is relatively *fast* and can be made faster by limiting the branches to open to the two branches *adjacent* to the branch closed in the branch-exchange mechanism
- The main drawback of the method is that it could easily stop into a *local minimum*
- The advantages are the relatively fast execution and the possibility of obtaining a feasible *trajectory* of operations that can be applied to the initial configuration to obtain the one provided by the algorithm

Simulated annealing (SA) algorithm

- Based on the *branch-exchange* method, initial *configuration* $X^{(0)} \in \boldsymbol{X}$
- Iterative process with *control parameter* c_m progressively decreased
- At the *m*-th iteration:
  - o initial configuration ⇒ *best* configuration found at the level c_{m-1}
  - o *random* extraction of an *open* branch to close (loop creation)
  - o *random* extraction (from the loop) of a *closed* branch to open
  - o *acceptance* of any configuration with *lower* objective function
  - o *conditional acceptance* of configurations with worse objective function
  - o worsening Δf depending on the acceptance *threshold* δ_p
  - o random number *r* extracted from a *uniform* distribution in [0,1]
  - o worsening Δf *accepted* if $e^{-\Delta f / c_m} > r$
  - o *stop* criteria (applied simultaneously):
    max number M_C of configurations *accepted* (active at “high”c_m);
    max number M_A of configurations *analysed* (active at “low”c_m);
- *Stop criterion*: the variation of the total losses does not exceed a specified tolerance for N_S successive control parameter levels

Objective function minimisation for SA

$$\min_{X \in \mathrm{X}} f(X) = \sum_{b \in \boldsymbol{B}} R_b I_b^2 \left[1 + \sum_{k \in \boldsymbol{K}} \rho_V \left(V_k^{\max} - V_k\right)^2 + \right.$$

$$+ \sum_{k \in \boldsymbol{K}} \rho_V \left(V_k - V_k^{MIN}\right)^2 + \sum_{k \in \boldsymbol{K}} \rho_{ef} \left(I_k^{ef,\max} - I_k^{ef}\right)^2 +$$

$$+ \sum_{k \in \boldsymbol{K}} \rho_{sc} \left(I_k^{sc,\max} - I_k^{sc}\right)^2 + \sum_{b \in \boldsymbol{B}} \rho_{th} \left(I_b^{th,\max} - I_b^{th}\right)^2 +$$

$$\left. + \sum_{k \in \boldsymbol{K}} \rho_{cb} \left(n_{ck}(path(k,s)) - 1\right) + \rho_{ds}\, n_{ds} + \rho_{op} \left(n - n^{\max}\right) \right]$$

The ρ coefficients are the *penalty factors* applied to the constraints and are null if the corresponding constrained variable does not exceed its limit

Parameters of the SA algorithm

- initial control parameter c_0
- max number M_A of configurations *analysed* at each control parameter
- max number M_C of configurations *accepted* at each control parameter
- *cooling rate* α (rate of decrease of the control parameter at iteration m given as $c_m = \alpha\, c_{m-1}$)
- acceptance *threshold* δ_p
- probability p_0 of accepting an objective function *worsening* (for the initialisation of the control parameter c_0)
- maximum successive iterations *without changes* Ns (for stop criterion)
- *seed* ξ for random numbers extraction (fixed to ensure repeatability of the results)

Initialisation of the control parameter

- The control parameter c_0 is initialised by calculating a given number N_0 of configurations with objective function *worse* than the one calculated in the base configuration
- The average worsening in the N_0 cases is $\Delta\bar{f}_p$
- Considering a *given probability* p_0 of accepting an objective function worsening, imposing the condition

$$e^{-\Delta\bar{f}_p / c_0} = p_0$$

it is possible to obtain $c_0 = \dfrac{\Delta\bar{f}_p}{\ln\left(1/p_0\right)}$

Interpretation of the acceptance condition

- A worsening Δf is *accepted* if $e^{-\Delta f / c_m} > r$
- The *acceptance* of a condition leading to the same worsening becomes *lower* when the iterative process continues
- Example with initial control parameter c_0 = 1, and *cooling rate* α = 0.98
- At the iteration m = 3:

  the cooling rate is c_3 = 1*0.98*0.98 = 0.9604

  for a worsening Δf = 0.1, $e^{-\Delta f / c_3}$ = 0.901
- At the iteration m = 5, the cooling rate is

  c_5 = 1*0.98*0.98*0.98*0.98 = 0.9224

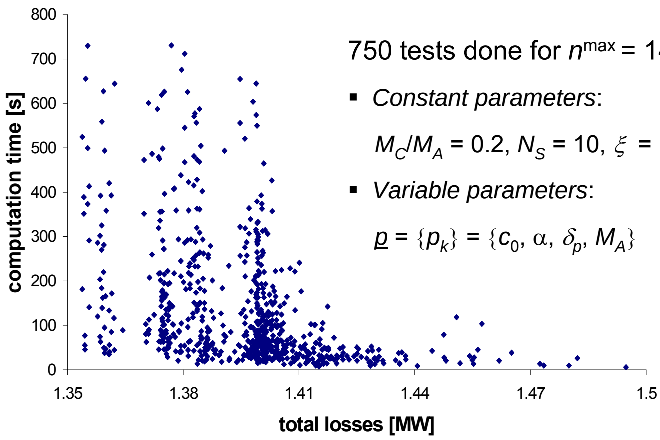
  for a worsening Δf = 0.1, $e^{-\Delta f / c_5}$ = 0.897
- If in both cases the random number extracted is r = 0.9, for m = 3 the configuration is accepted and for m = 5 is not accepted

Convergence of the SA algorithm

- bi-asymptotic theoretical convergence to the global optimum
  - √ theoretical proof has been given of the convergence of the algorithm to the global optimum for infinite iterations at decreasing control parameter levels, with infinite iterations at each control parameter level
- high computational burden
  - √ relatively long computation time for a large distribution system
  - √ a set of tests could take some hours on a Pentium IV personal computer
- high sensitivity to the parameters of the algorithm
  - √ no correlation has been found between the parameters of the algorithm and the objective function

Example on a MV system

- Good objective functions could be obtained from *relatively fast* executions (e.g., relatively *low* initial values of the control parameter)



750 tests done for n^{max} = 14 with

- *Constant parameters*:

  M_C/M_A = 0.2, N_S = 10, ξ = 1
- *Variable parameters*:

  $\underline{p} = \{p_k\} = \{c_0, \alpha, \delta_p, M_A\}$

Comments on the SA algorithm

- The SA-based algorithm could give low-losses solutions, *but*:
  - it is not possible to clearly identify *a priori* how to fix the parameters of the algorithm to reach a solution with relatively low total losses
  - very *poor correlations* have been computed between the main parameters of the algorithm and the value of the objective function
  - higher computation times do *not* imply better solutions are reached
  - solutions with very *low total losses* have been obtained even by using a combination of parameters leading to fast executions
  - a "*one-shot*" simulation, performed by assigning the parameters of the algorithm with some predefined criterion, does not guarantee the effectiveness of the solution
  - if the computation time is not a major issue, running the SA-based algorithm *several times* with parameters set to provide relatively fast solutions is suggested

Comparisons between IT and SA

- IT is (2 orders of magnitude) faster than the SA
- The SA best solutions are slightly better than the IT ones