OPTIMIZATION SOLUTIONS

LARGE-SCALE OPTIMAL POWER FLOW WITH NO GUARANTEE ON FEASIBILITY

SYLVAIN MOURET, A. RENAUD, M. RUIZ, P. GIRARDEAU - ARTELYS
J. MAEGHT, S. FLISCOUNAKIS, P. PANCIATICI - RTE

June 2013

www.artelys.com

CONTENTS

- General framework
  - iTesla, large-scale OPF models with no guarantee on feasibility, type of model (intensity limits, phase-shifting transformers)
  - Numerical experiments on real data from European TSOs
- Problems encountered by solving a direct approach
  - Difficulties to converge
  - Not possible to know the status of the solution and characteristics of the network state
- Proposed solution: a progressive filtering process
  - The direct approach is replaced by a multi-step solution process
  - Each step amounts to solving an easier problem
- Computational experiments

Artelys | OPTIMIZATION SOLUTIONS

GENERAL FRAMEWORK

- iTesla is a pan-European R&D project that aims at assessing the security of a large scale power network by means of security rules computed offline
  - Coordinated by RTE (Réseau de Transport d'Electricité)
  - Includes 6 European TSOs and 13 R&D companies
  - Official website: http://www.itesla-project.eu/

- Two major platforms developed
  1. **Offline**: explore the network state space to draw the separation between stable and unstable states (using data mining techniques)
  2. **Online**: evaluate computed security rules on the current network situation and provide recommendations to TSOs

THE ITESLA PROJECT

Offline platform

- Sampling of network states
- Infeasible states detected through steady-state optimization
- Unstable states detected through dynamic optimization
- Data mining on the results

Every week

Online platform

- Data acquisition from European TSOs (24 hour forecasts)
- Data merging
- Security assessment
- Recommendation for the operators

Every day

THE OPF MODEL

- Here we focus on the offline task
  - Monte Carlo simulations provide us with many network states (~10,000)
  - We want to filter out the ones that are not feasible
- The mathematical model is a modified AC-OPF
  - Polar PQV formulation
  - Limits on voltage magnitudes
  - Maximum intensity levels on lines (nonlinear inequality constraints)
  - Limits on production levels
  - Kirchhoff law at each node (nonlinear equality constraints)

SPECIFIC FEATURES

- When necessary, fixed injection can be modified
  - Positive fixed injection at a node can be decreased
    - Production curtailment of fatal production unit (***PC***)
  - Negative fixed injection at a node can be increased
    - Load shedding (***LS***)
- Use of specific absolute tolerance on each constraint
  - Limits on voltage magnitude
  - Maximum limit on intensity level
  - Balance of active and reactive power at each node
  - Limits on active and reactive level of production units

DESCRIPTION OF THE DATASET

- Network data comes from real data (recollection of network data from several European TSOs)
  - $>$ 7000 nodes
  - $>$ 8000 lines
  - $\sim$ 700 production units
- This leads to a large scale nonlinear optimization problem
  - The input data has not been verified
  - We have no guarantee that a feasible solution actually exists
- The dataset is composed of 843 test cases which correspond to a whole week of real data from European TSOs

OBJECTIVES

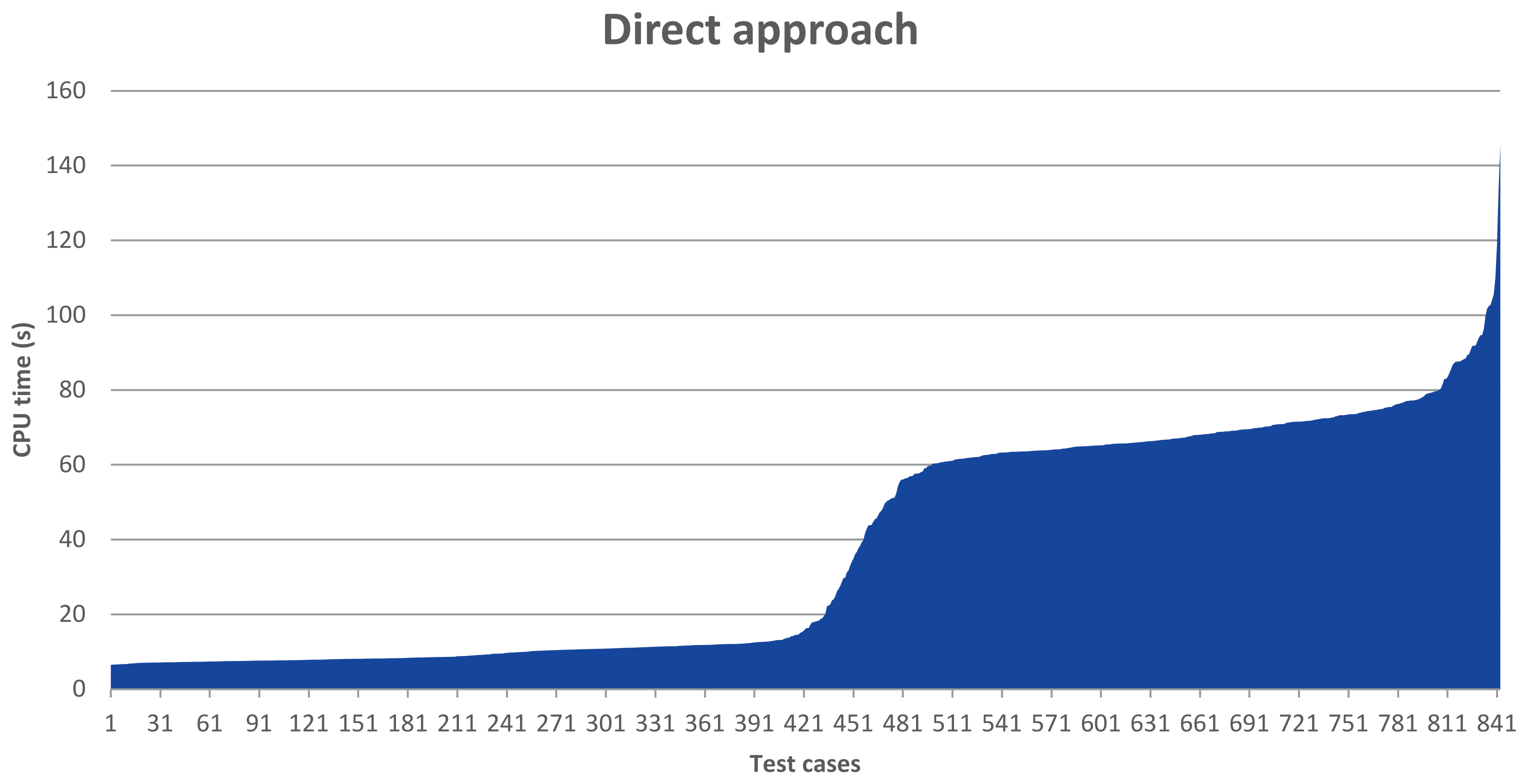
- The goal is to answer the following questions
  - Is the OPF model feasible without ***PC*** or ***LS***?
  - Can the OPF model be made feasible with only ***PC***?
  - Can the OPF model be made with both ***PC*** and ***LS***?
- If no ***LS*** is needed, ***PC*** is used as little as possible
- If needed, ***LS*** is used as little as possible, even if this leads to use more ***PC***

DIRECT APPROACH

DIRECT APPROACH

- The objective is to minimize load shedding and production curtailment on each node
  - $\min \boldsymbol{LS} + 0.1 \cdot \boldsymbol{PC}$
  - Reminding that:
    - If no ***LS*** is needed, ***PC*** is used as little as possible
    - If needed, ***LS*** is used as little as possible, even if this leads to use more ***PC***
- The problem is solved directly using
  - KNITRO 8.1.1, a state-of-the-art nonlinear optimization solver
  - AMPL, a standard modeling language for mathematical optimization
- KNITRO uses an interior-point method to solve the OPF
  - Newton-Raphson + line search descent, projected conjugate gradient, etc.
  - The number of interior-point iterations is limited to 200

COMPUTATIONAL RESULTS



- Out of 843 test cases
  - 360 test cases reached the iteration limit
- The feasibility assessment is based on the last solution iterate

REMARKS ON THE DIRECT APPROACH

- When the maximal number of iterations is reached, no conclusion can be made on the test case
  - The solution point may be infeasible while the test case actually is feasible
  - The solution point may be feasible with positive ***PC*** or ***LS***, while a solution with no ***PC*** or ***LS*** actually exists (and we would like to find it)
- If the test case if found infeasible within the iteration limit, the origin of the infeasibility remains unclear

PROGRESSIVE FILTERING APPROACH

PROGRESSIVE FILTERING

- A progressive filtering approach has been developed to achieve the following goals:
  - **gain stability in terms of convergence** and CPU usage
  - **obtain more detailed information on the reasons why a network state is infeasible**:
    - Can we make it feasible by curtailing some production at specific network nodes?
    - Is it necessary to perform load shedding as well?
    - In which nodes should the power injection be modified?

PROGRESSIVE FILTERING

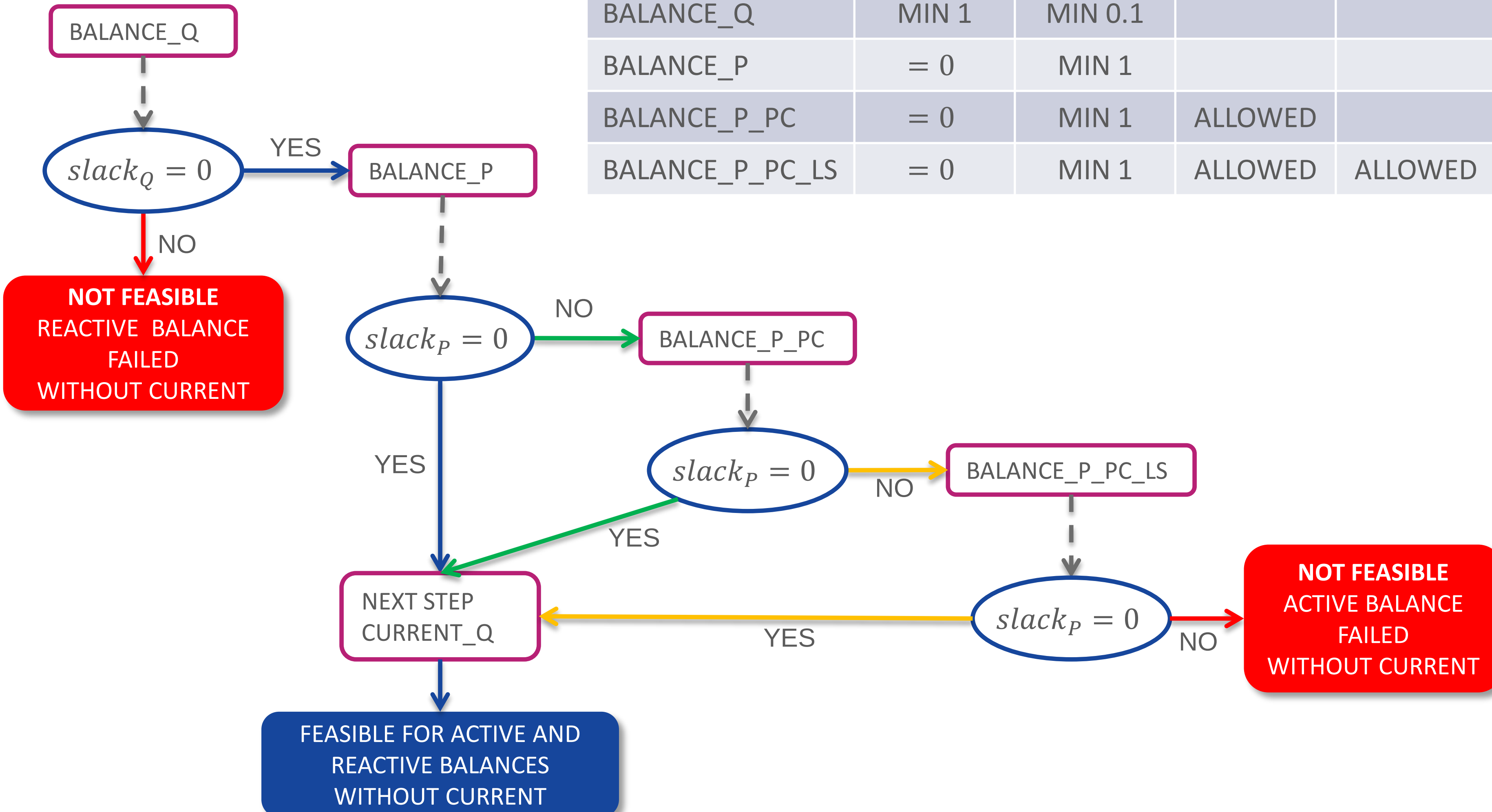
- The maximum limits on intensity levels make the problem much harder to solve
  - Main reason: they act as a capacity constraint on line power flows
  - All models solve within 10 seconds without such limits
    - Production targets and demands are usually well balanced
- Relaxing the power balance constraints tends to decrease the power flow needed on lines
  - This tends to decrease the current intensity level : $|I|^2 = \frac{|S|^2}{|V|^2}$
- Thus, slack variables are applied to active and reactive power balances only, as it is sufficient to make the model feasible
  - $slack_P$ on active power balances
  - $slack_Q$ on reactive power balances

DESCRIPTION OF THE APPROACH

- The progressive filtering approach is applied twice
  - without current intensity levels
  - with current intensity levels
- Each step of the filtering procedure has a dedicated objective function and may or may not use slack variables
- Each problem must be solved within less than 100 iterations
- The localization of ***PC*** or ***LS*** is only perform when the maximum limits on intensity level are enforced
  - Intensity limits have a great impact of the location of ***PC*** and ***LS***

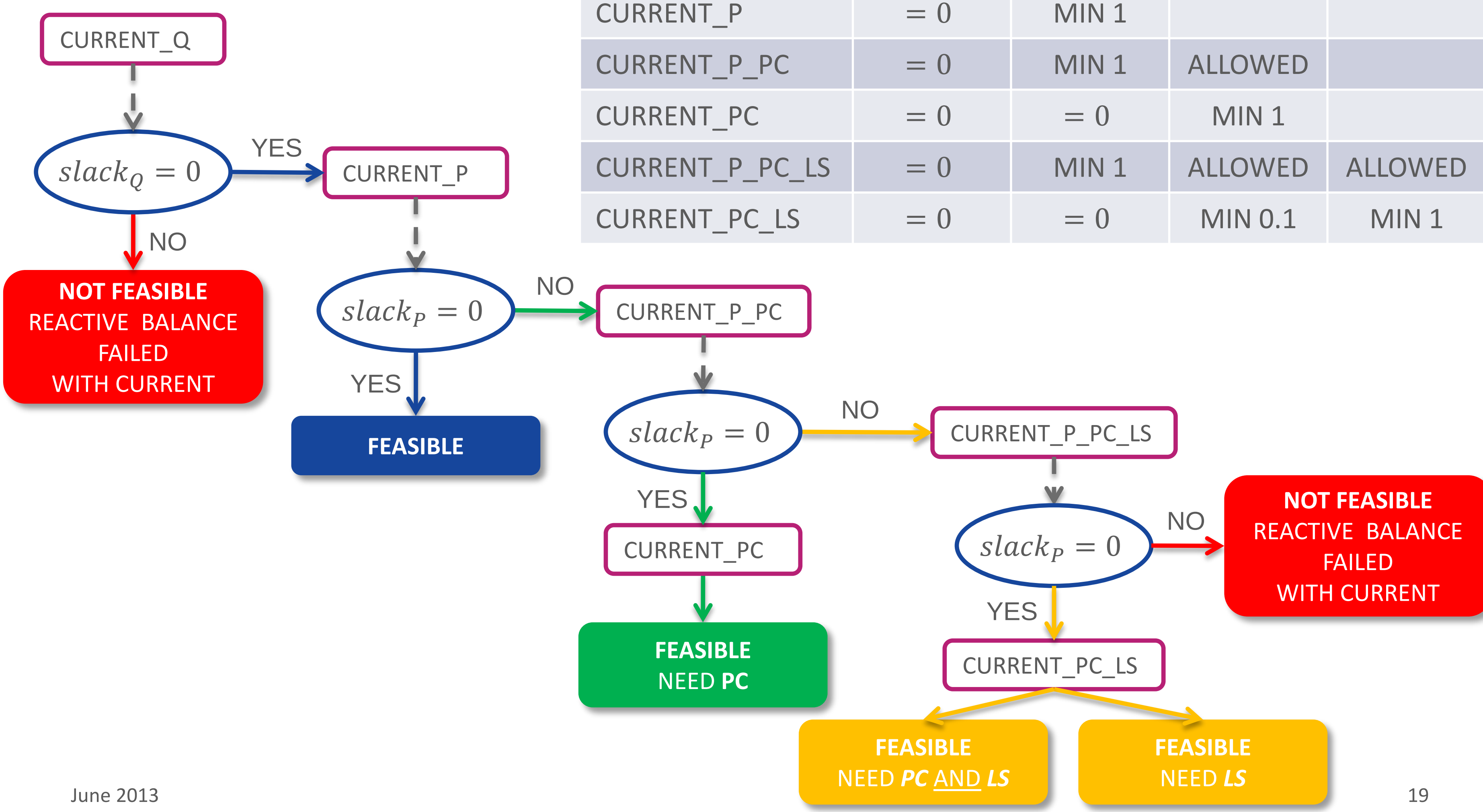
STEPS WITHOUT INTENSITY LIMIT

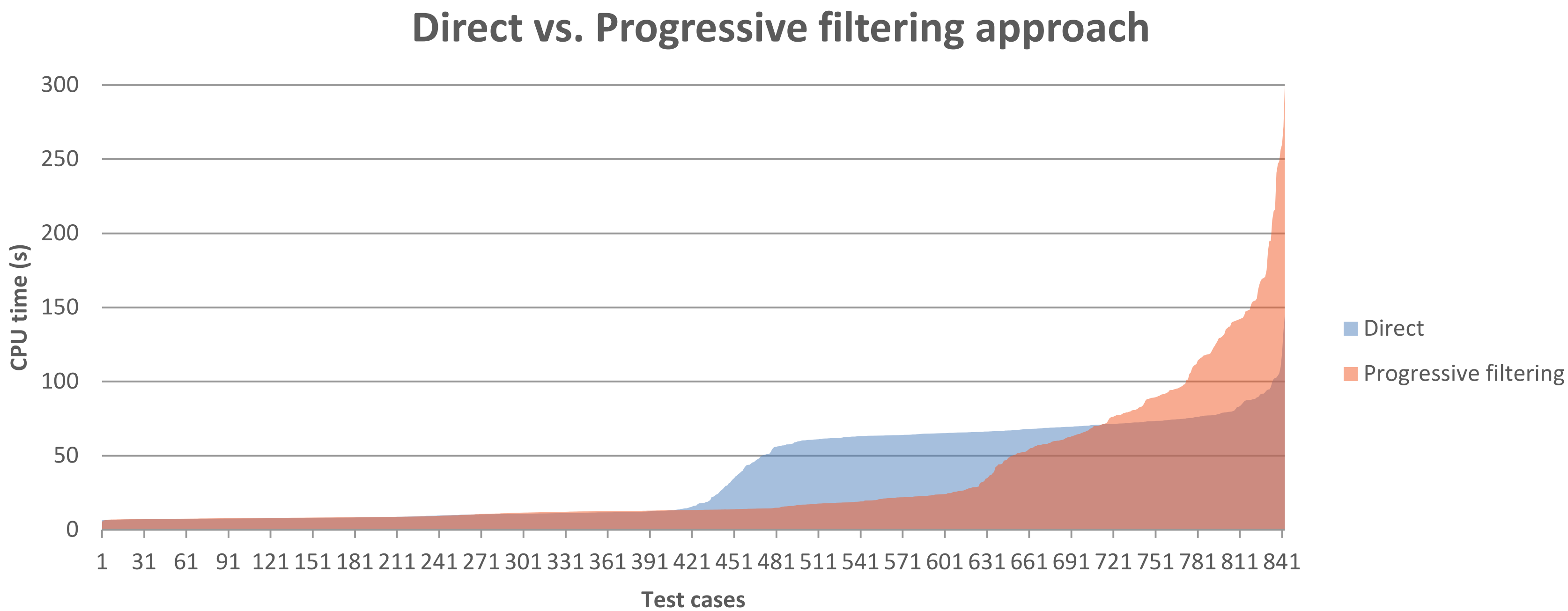
STEP	$slack_Q$	$slack_P$	PC	LS
BALANCE_Q	MIN 1	MIN 0.1		
BALANCE_P	$= 0$	MIN 1		
BALANCE_P_PC	$= 0$	MIN 1	ALLOWED	
BALANCE_P_PC_LS	$= 0$	MIN 1	ALLOWED	ALLOWED



STEPS WITH INTENSITY LIMIT

STEP	$slack_Q$	$slack_P$	*PC*	*LS*
CURRENT_Q	MIN 1	MIN 0.1		
CURRENT_P	$= 0$	MIN 1		
CURRENT_P_PC	$= 0$	MIN 1	ALLOWED	
CURRENT_PC	$= 0$	$= 0$	MIN 1	
CURRENT_P_PC_LS	$= 0$	MIN 1	ALLOWED	ALLOWED
CURRENT_PC_LS	$= 0$	$= 0$	MIN 0.1	MIN 1

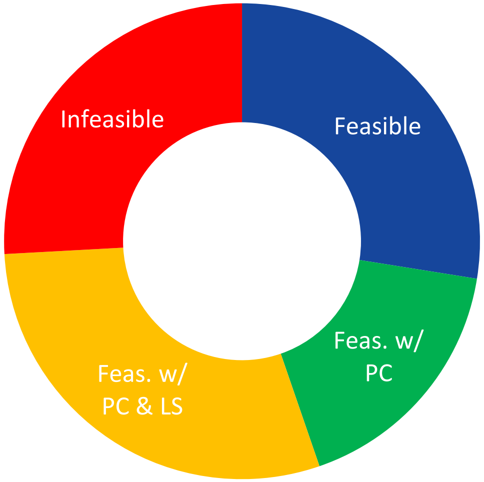




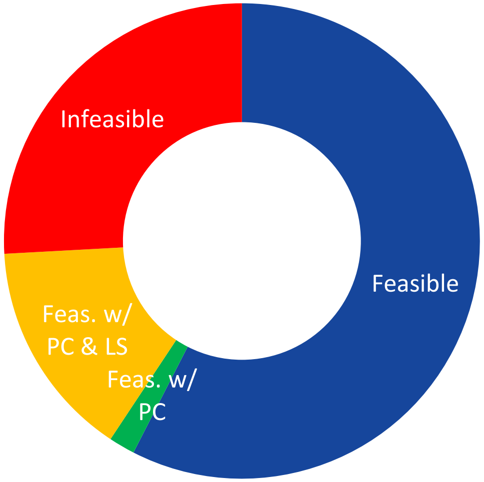
- About 50 minutes of total CPU time is saved over the 843 test cases
  - Direct approach: 8 hours 47 minutes
  - Progressive filtering approach: 7 hours 58 minutes
- However, some test cases (not feasible without ***PC*** or ***LS***) are solved with high CPU times
  - More time is spent in order to recover detailed infeasibility information

SOLUTION STATUSES

Direct approach Solution status



Progressive filtering Solution status



The direct approach missed:

- 253 instances that are found feasible without ***PC***
- 123 instances that are found feasible with or without ***PC***

The progressive approach provides more information on infeasibilities:

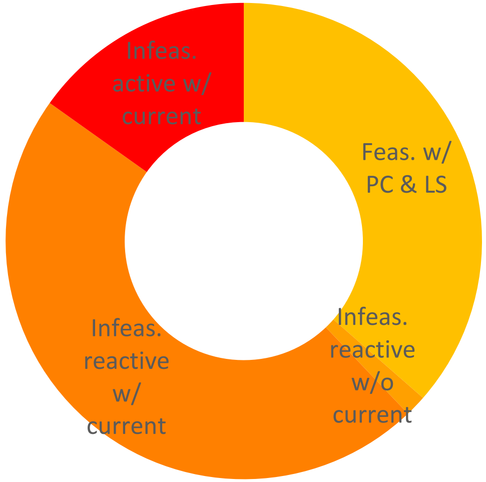
- issues with active/reactive power balance, issues with intensity limits
- localization of such difficulties

SOLUTION STATUSES

- Reasons for infeasibilities:
  - About 1/3 of infeasible models can be made feasible by using ***LS***
  - About 1/2 of infeasibilities are due to reactive power balance issues
  - About 1/6 of infeasibilities are due to active power balance issues

- The average CPU time per step is
  - 4.5 seconds for power balance slack minimization without intensity limits
  - 13.5 seconds for power balance slack minimization with intensity limits
  - 32.0 seconds for ***PC*** minimization (when used)
  - 44.4 seconds for ***LS*** minimization (when used)

Progressive filtering
Solution status

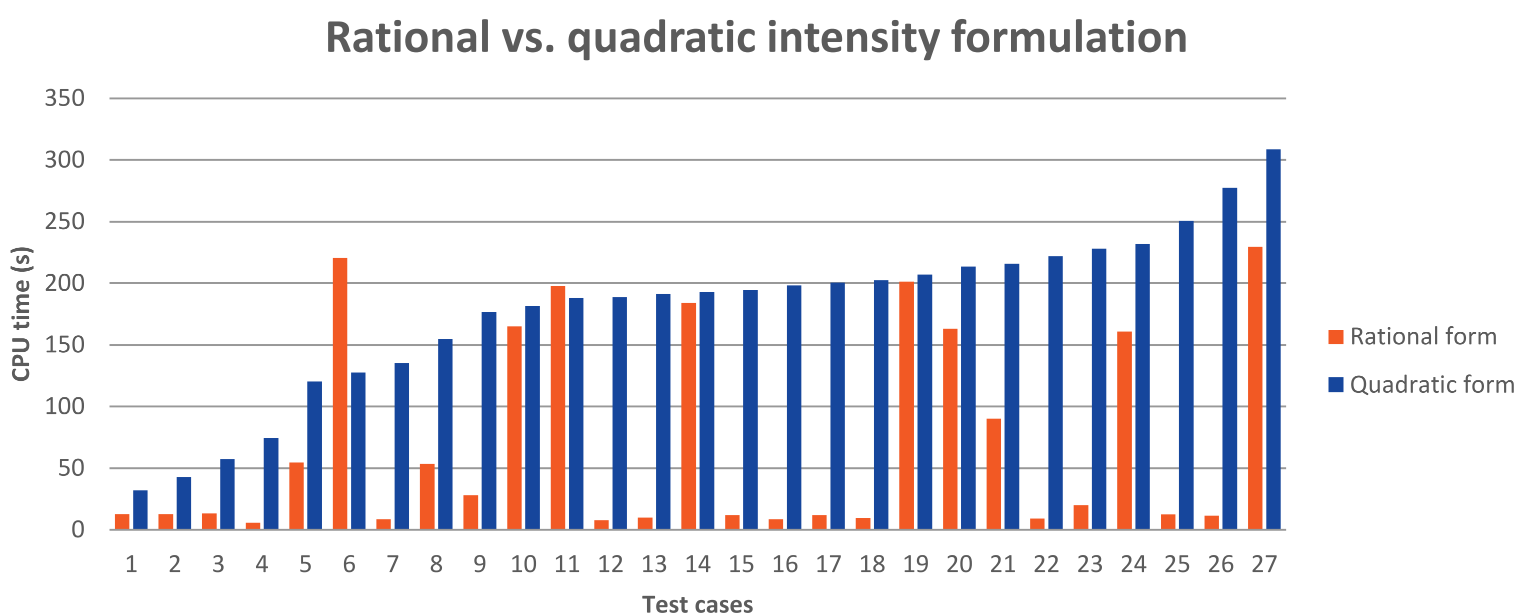


Artelys | OPTIMIZATION SOLUTIONS

ON THE INTENSITY LIMIT

MAXIMUM INTENSITY LEVEL

- The maximum intensity level constraint can be expressed in a quadratic form or in a rational form
  - $|I|^2 \leq \bar{I}^2$ or $\sqrt{|I|^2} \leq \bar{I}$
  - The rational formulation scales better and leads to better performance
    - Demonstrated by an experiment on a reduced dataset of 27 test cases



CONCLUSION

CONCLUSION

- A progressive filtering procedure has been developed in order to detect infeasibilities for large-scale OPF problems
- The procedure is tested on a whole week of real data from European TSOs (843 test cases)
  - The filtering process is able to solve more instances than the direct approach
- The KNITRO performance was greatly improved by
  - scaling the model
  - using constraint-specific feasibility tolerances
    - avoids unnecessary long convergence runs to achieve default tolerances
    - new feature that will be available in the next KNITRO release

THANK YOU FOR YOUR ATTENTION

ANY QUESTIONS?

MODELING OF THE LINES