A two-step sequential linear programming algorithm for MINLP problems:

An application to gas transmission networks

Julio González-Díaz
Ángel M. González-Rueda
María P. Fernández de Córdoba

University of Santiago de Compostela
Technological Institute for Industrial Mathematics (ITMATI)

....

RSME2017
2ARA602A

February 3rd, 2017

A two-step sequential linear programming algorithm for MINLP problems:

An application to gas transmission networks

GANESO$^{\mathrm{TM}}$: Gas Networks Simulation and Optimization

GANESO$^{\mathrm{TM}}$: **Ga**s **Ne**tworks **S**imulation and **O**ptimization

- **GANESO**$^{\mathrm{TM}}$ is a software developed by researchers at **USC** and **ITMATI** for **Reganosa Company**

GANESO$^{\mathrm{TM}}$: Gas Networks Simulation and Optimization

- **GANESO**$^{\mathrm{TM}}$ is a software developed by researchers at **USC** and **ITMATI** for **Reganosa Company**
- Ongoing project that started in **2011**; around 15 researchers have participated.

GANESO$^{\mathrm{TM}}$: Gas Networks Simulation and Optimization

- **GANESO**$^{\mathrm{TM}}$ is a software developed by researchers at **USC** and **ITMATI** for **Reganosa Company**
- Ongoing project that started in **2011**; around 15 researchers have participated. More than **600.000 €** invested by Reganosa

GANESO$^{\mathrm{TM}}$: Gas Networks Simulation and Optimization

- **GANESO**$^{\mathrm{TM}}$ is a software developed by researchers at **USC** and **ITMATI** for **Reganosa Company**
- Ongoing project that started in **2011**; around 15 researchers have participated. More than **600.000 €** invested by Reganosa
- Main functionalities of **GANESO**$^{\mathrm{TM}}$:
  - Steady-state and transient simulation
    - Gas loss analysis
    - Gas quality tracking
    - Linepack control
  - Steady-state optimization
    - Network planning and design under uncertainty
  - Computation of tariffs for network access
  - Database management for indexing network scenarios
  - User Interface (daily usage of GANESO$^{\mathrm{TM}}$ by end-user)

GANESO$^{\mathrm{TM}}$: Gas Networks Simulation and Optimization

- **GANESO**$^{\mathrm{TM}}$ is a software developed by researchers at **USC** and **ITMATI** for **Reganosa Company**
- Ongoing project that started in **2011**; around 15 researchers have participated. More than **600.000 €** invested by Reganosa
- Main functionalities of **GANESO**$^{\mathrm{TM}}$:
  - Steady-state and transient simulation
    - Gas loss analysis
    - Gas quality tracking
    - Linepack control
  - Steady-state optimization
    - Network planning and design under uncertainty
  - Computation of tariffs for network access
  - Database management for indexing network scenarios
  - User Interface (daily usage of GANESO$^{\mathrm{TM}}$ by end-user)

GANESO$^{\mathrm{TM}}$: Gas Networks Simulation and Optimization

- **GANESO**$^{\mathrm{TM}}$ is a software developed by researchers at **USC** and **ITMATI** for **Reganosa Company**
- Ongoing project that started in **2011**; around 15 researchers have participated. More than **600.000 €** invested by Reganosa
- Main functionalities of **GANESO**$^{\mathrm{TM}}$:
  - Steady-state and transient simulation
    - Gas loss analysis
    - Gas quality tracking
    - Linepack control
  - Steady-state optimization
    - Network planning and design under uncertainty
  - Computation of tariffs for network access
  - Database management for indexing network scenarios
  - User Interface (daily usage of GANESO$^{\mathrm{TM}}$ by end-user)

Nonlinear optimization

GANESO^{TM}: Gas Networks Simulation and Optimization

- **GANESO**^{TM} is a software developed by researchers at **USC** and **ITMATI** for **Reganosa Company**
- Ongoing project that started in **2011**; around 15 researchers have participated. More than **600.000 €** invested by Reganosa
- Main functionalities of **GANESO**^{TM}:
    - Steady-state and transient simulation
        - Gas loss analysis
        - Gas quality tracking
        - Linepack control
    - Steady-state optimization
        - Network planning and design under uncertainty
    - Computation of tariffs for network access
    - Database management for indexing network scenarios
    - User Interface (daily usage of GANESO^{TM} by end-user)

Nonlinear optimization

GANESO$^{\mathrm{TM}}$: Gas Networks Simulation and Optimization

- **GANESO**$^{\mathrm{TM}}$ is a software developed by researchers at **USC** and **ITMATI** for **Reganosa Company**
- Ongoing project that started in **2011**; around 15 researchers have participated. More than **600.000 €** invested by Reganosa
- Main functionalities of **GANESO**$^{\mathrm{TM}}$:
  - Steady-state and transient simulation
    - Gas loss analysis
    - Gas quality tracking
    - Linepack control
  - Steady-state optimization
    - Network planning and design under uncertainty
  - Computation of tariffs for network access
  - Database management for indexing network scenarios
  - User Interface (daily usage of GANESO$^{\mathrm{TM}}$ by end-user)

Nonlinear optimization

Stochastic programming

GANESO$^{\mathrm{TM}}$: Gas Networks Simulation and Optimization

- **GANESO**$^{\mathrm{TM}}$ is a software developed by researchers at **USC** and **ITMATI** for **Reganosa Company**
- Ongoing project that started in **2011**; around 15 researchers have participated. More than **600.000 €** invested by Reganosa
- Main functionalities of **GANESO**$^{\mathrm{TM}}$:
  - Steady-state and transient simulation
    - Gas loss analysis
    - Gas quality tracking
    - Linepack control
  - Steady-state optimization
    - Network planning and design under uncertainty
  - Computation of tariffs for network access
  - Database management for indexing network scenarios
  - User Interface (daily usage of GANESO$^{\mathrm{TM}}$ by end-user)

Nonlinear optimization

Gas transmission networks

Gas transmission networks

Gas transmission networks

Gas transmission networks

Ingredients of the optimization problem

Ingredients of the optimization problem

Identify feasible gas flows

Ingredients of the optimization problem

Identify feasible gas flows

Main problem constraints

- Meet demands (security of supply)
- Gas pressure is kept within specified bounds

Ingredients of the optimization problem

Identify feasible gas flows

Main problem constraints

- Meet demands (security of supply)
- Gas pressure is kept within specified bounds

Ingredients of the optimization problem

Identify feasible gas flows

Main problem constraints

- Meet demands (security of supply)
- Gas pressure is kept within specified bounds

Different objective functions

- Minimize gas consumption at compressor stations
- Minimize boil-off gas at regasification plants
- Maximize network linepack
- Maximize/minimize exports of different zones
- Control bottlenecks

Ingredients of the optimization problem

Identify feasible gas flows

Main problem constraints

- Meet demands (security of supply)
- Gas pressure is kept within specified bounds

Different objective functions

- Minimize gas consumption at compressor stations
- Minimize boil-off gas at regasification plants
- Maximize network linepack
- Maximize/minimize exports of different zones
- Control bottlenecks

Network flow problem

Network flow problem

Flow conservation constraints

$$\sum_{k \in A_i^{\text{ini}}} q_k - \sum_{k \in A_i^{\text{fin}}} q_k = c_i$$

$\forall i \in N^C$ demand nodes

$$0 \leq \sum_{k \in A_i^{\text{ini}}} q_k - \sum_{k \in A_i^{\text{fin}}} q_k \leq s_i$$

$\forall i \in N^S$ supply nodes

Network flow problem

Flow conservation constraints

$$\sum_{k \in A_i^{\text{ini}}} q_k - \sum_{k \in A_i^{\text{fin}}} q_k = c_i$$

$\forall i \in N^C$ demand nodes

$$0 \leq \sum_{k \in A_i^{\text{ini}}} q_k - \sum_{k \in A_i^{\text{fin}}} q_k \leq s_i$$

$\forall i \in N^S$ supply nodes

Box Constraints

$$\underline{q}_k \leq q_k \leq \bar{q}_k$$

$\forall k \in A$ flow bounds

Network flow problem

Flow conservation constraints

$$\sum_{k \in A_i^{\text{ini}}} q_k - \sum_{k \in A_i^{\text{fin}}} q_k = c_i$$

$\forall i \in N^C$ demand nodes

$$0 \leq \sum_{k \in A_i^{\text{ini}}} q_k - \sum_{k \in A_i^{\text{fin}}} q_k \leq s_i$$

$\forall i \in N^S$ supply nodes

Box Constraints

$$\underline{q}_k \leq q_k \leq \bar{q}_k$$

$\forall k \in A$ flow bounds

$$\underline{p}_i^2 \leq {p_i}^2 \leq \bar{p}_i^2$$

$\forall i \in N$ **pressure bounds**

Network flow problem

Flow conservation constraints

$$\sum_{k\in A_i^{\text{ini}}} q_k - \sum_{k\in A_i^{\text{fin}}} q_k = c_i$$

$\forall i \in N^C$ demand nodes

$$0 \le \sum_{k\in A_i^{\text{ini}}} q_k - \sum_{k\in A_i^{\text{fin}}} q_k \le s_i$$

$\forall i \in N^S$ supply nodes

Box Constraints

$$\underline{q}_k \le q_k \le \bar{q}_k$$

$\forall k \in A$ flow bounds

$$\underline{p}_i^2 \le {p_i}^2 \le \bar{p}_i^2$$

$\forall i \in N$ **pressure bounds**

Variables of the optimization problem

- Flow through each pipe
- **Pressure** at each node

Gass loss equations

Given a pipe between two nodes i and j, we have

$$p_i{}^2 - p_j{}^2 = \frac{16 L_k \lambda_k}{\pi^2 D_k^5} Z(p_m, T_m) R T_m |q_{ij}| q_{ij} + \frac{2g}{R T_m} \frac{p_i{}^2 + p_j{}^2}{2 Z(p_m, T_m)} (h_j - h_i)$$

Gass loss equations

Given a pipe between two nodes i and j, we have

$$p_i{}^2 - p_j{}^2 = \frac{16 L_k \lambda_k}{\pi^2 D_k^5} Z(p_m, T_m) R T_m |q_{ij}| q_{ij} + \frac{2g}{R T_m} \frac{p_i{}^2 + p_j{}^2}{2 Z(p_m, T_m)} (h_j - h_i)$$

Gass loss equations

Given a pipe between two nodes i and j, we have

$$p_i{}^2 - p_j{}^2 = \frac{16 L_k \lambda_k}{\pi^2 D_k^5} Z(p_m, T_m) R T_m |q_{ij}| q_{ij} + \frac{2g}{R T_m} \frac{p_i{}^2 + p_j{}^2}{2Z(p_m, T_m)} (h_j - h_i)$$

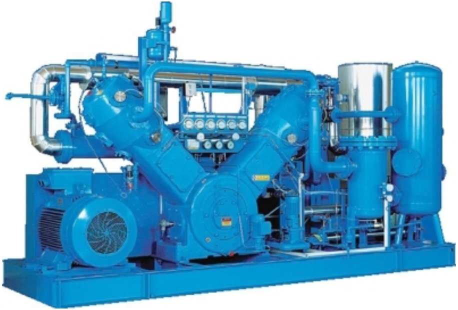
Gass loss equations

Given a pipe between two nodes i and j, we have

$$p_i{}^2 - p_j{}^2 = \frac{16 L_k \lambda_k}{\pi^2 D_k^5} Z(p_m, T_m) R T_m |q_{ij}| q_{ij} + \frac{2g}{R T_m} \frac{p_i{}^2 + p_j{}^2}{2 Z(p_m, T_m)} (h_j - h_i)$$

As many nonlinear constraints as pipes

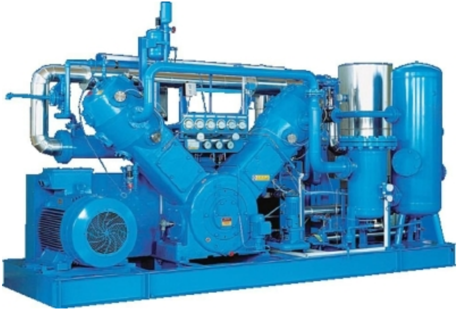
Gas consumption at compressors



Given input pressure p_i and output pressure p_j, we have

$$g_{ij} = \frac{1}{e_h H^c} \frac{\gamma}{\gamma - 1} Z(p_m, T_{in}) R T_{in} \left(\left(\frac{p_j}{p_i}\right)^{\frac{\gamma-1}{\gamma}} - 1 \right) q_{ij}$$

Gas consumption at compressors



Given input pressure p_i and output pressure p_j, we have

$$g_{ij} = \frac{1}{e_h H^c} \frac{\gamma}{\gamma - 1} Z(p_m, T_{in}) R T_{in} \left(\left(\frac{p_j}{p_i}\right)^{\frac{\gamma-1}{\gamma}} - 1 \right) q_{ij}$$

As many nonlinear constraints as compressors in the network

Nonlinear nonconvex optimization problem

Nonlinear nonconvex optimization problem

Obj. Function: $\min \sum_{k \in A^c} g_k$

Nonlinear nonconvex optimization problem

Obj. Function: $\min \sum_{k \in A^c} g_k$

Box Constraints

$$\underline{p}_i^2 \leq {p_i}^2 \leq \bar{p}_i^2 \qquad \forall i \in N \text{ pressure bounds}$$

$$\underline{q}_k \leq q_k \leq \bar{q}_k \qquad \forall k \in A \text{ flow bounds}$$

Flow conservation constraints

$$\sum_{k \in A_i^{\text{ini}}} q_k - \sum_{k \in A_i^{\text{fin}}} q_k = c_i \qquad \forall i \in N^C \text{ flow conservation at demand nodes}$$

$$0 \leq \sum_{k \in A_i^{\text{ini}}} q_k - \sum_{k \in A_i^{\text{fin}}} q_k \leq s_i \qquad \forall i \in N^S \text{ flow conservation at supply nodes}$$

Gas loss constraints

$${p_i}^2 - {p_j}^2 = \frac{16 L_k \lambda_k}{\pi^2 D_k^5} Z(p_m, T_m) R T_m |q_k| q_k + \qquad \forall k \in A^n \text{ gas loss } (\lambda_k \text{ Weymouth})$$

$$+ \frac{2g}{RT_m} \frac{{p_i}^2 + {p_j}^2}{2Z(p_m, T_m)} (h_j - h_i) \qquad \text{height difference term}$$

Gas consumption constraints

$$g_k = \frac{1}{e_h H^c} \frac{\gamma}{\gamma - 1} Z(p_m, T_{in}) R T_{in} \left(\left(\frac{p_j}{p_i}\right)^{\frac{\gamma-1}{\gamma}} - 1 \right) q_k$$

Nonlinear nonconvex optimization problem (continuous)

Obj. Function: $\min \sum_{k \in A^c} g_k$

Box Constraints

$\underline{p}_i^2 \leq p_i{}^2 \leq \bar{p}_i^2$ $\quad \forall i \in N$ pressure bounds

$\underline{q}_k \leq q_k \leq \bar{q}_k$ $\quad \forall k \in A$ flow bounds

Flow conservation constraints

$$\sum_{k \in A_i^{\text{ini}}} q_k - \sum_{k \in A_i^{\text{fin}}} q_k = c_i \qquad \forall i \in N^C \text{ flow conservation at demand nodes}$$

$$0 \leq \sum_{k \in A_i^{\text{ini}}} q_k - \sum_{k \in A_i^{\text{fin}}} q_k \leq s_i \qquad \forall i \in N^S \text{ flow conservation at supply nodes}$$

Gas loss constraints

$$p_i{}^2 - p_j{}^2 = \frac{16 L_k \lambda_k}{\pi^2 D_k^5} Z(p_m, T_m) R T_m |q_k| q_k + \qquad \forall k \in A^n \text{ gas loss } (\lambda_k \text{ Weymouth})$$

$$+ \frac{2g}{R T_m} \frac{p_i{}^2 + p_j{}^2}{2 Z(p_m, T_m)} (h_j - h_i) \qquad \text{height difference term}$$

Gas consumption constraints

$$g_k = \frac{1}{e_h H^c} \frac{\gamma}{\gamma - 1} Z(p_m, T_{in}) R T_{in} \left(\left(\frac{p_j}{p_i}\right)^{\frac{\gamma-1}{\gamma}} - 1 \right) q_k$$

Size and complexity of real instances

Size and complexity of real instances

Spanish primary gas network

- ≈ 1000 variables (≈ 500 pipes and ≈ 500 nodes)
- ≈ 1000 constraints (and ≈ 2000 box constraints)
- ≈ 500 constraints are **nonlinear**

Size and complexity of real instances

Spanish primary gas network

- ≈ 1000 variables (≈ 500 pipes and ≈ 500 nodes)
- ≈ 1000 constraints (and ≈ 2000 box constraints)
- ≈ 500 constraints are **nonlinear**

To be solved routinely by the company

(A twist on) Sequential Linear Programming Algorithms

Approaches to solve the problem

Spanish primary gas network

- ≈ 1000 variables (≈ 500 pipes and ≈ 500 nodes)
- ≈ 1000 constraints (and ≈ 2000 box constraints)
- ≈ 500 constraints are **nonlinear**

Approaches to solve the problem

Spanish primary gas network

- ≈ 1000 variables (≈ 500 pipes and ≈ 500 nodes)
- ≈ 1000 constraints (and ≈ 2000 box constraints)
- ≈ 500 constraints are **nonlinear**

How to solve this problem?

Approaches to solve the problem

Spanish primary gas network

- ≈ 1000 variables (≈ 500 pipes and ≈ 500 nodes)
- ≈ 1000 constraints (and ≈ 2000 box constraints)
- ≈ 500 constraints are **nonlinear**

How to solve this problem?

- **Global optimization algorithms** on approximations of the problem

  nonlinearities $\Longrightarrow$ piecewise linear functions $+$ integer variables

Approaches to solve the problem

Spanish primary gas network

- ≈ 1000 variables (≈ 500 pipes and ≈ 500 nodes)
- ≈ 1000 constraints (and ≈ 2000 box constraints)
- ≈ 500 constraints are **nonlinear**

How to solve this problem?

- **Global optimization algorithms** on approximations of the problem (cannot handle real-size problems)

  nonlinearities $\Longrightarrow$ piecewise linear functions $+$ integer variables

Approaches to solve the problem

Spanish primary gas network

- ≈ 1000 variables (≈ 500 pipes and ≈ 500 nodes)
- ≈ 1000 constraints (and ≈ 2000 box constraints)
- ≈ 500 constraints are **nonlinear**

How to solve this problem?

- **Global optimization algorithms** on approximations of the problem (cannot handle real-size problems)
  nonlinearities $\Longrightarrow$ piecewise linear functions $+$ integer variables
- **Local optimization algorithms** such as sequential linear programming, **SLP**, or sequential quadratic programming, **SQP**

Our initial approach

Classic SLP

Our initial approach

Classic SLP

- We get a solution using **Classic SLP**

Our initial approach

Classic SLP + Control Theory

- We get a solution using **Classic SLP**
- We refine it using **control theory** by including some second order elements

Our initial approach

Classic SLP + Control Theory

- We get a solution using **Classic SLP**
- We refine it using **control theory** by including some second order elements

Nothing specially original so far

Our initial approach

Classic SLP

- We get a solution using **Classic SLP**

Nothing specially original so far

Additional network elements

Elements that require the use of binary variables

Additional network elements

Elements that require the use of binary variables

- Different types of **control valves**
- **Operational ranges** of each compressor station
- Boil-off gas at regasification plants

Additional network elements

Elements that require the use of binary variables

- Different types of **control valves**
- **Operational ranges** of each compressor station
- Boil-off gas at regasification plants

Mixed-integer nonlinear nonconvex programming problem

- $\approx$ 1000 continuous variables and 1000 constraints
- No more than 100-200 binary variables

Additional network elements

Elements that require the use of binary variables

- Different types of **control valves**
- **Operational ranges** of each compressor station
- Boil-off gas at regasification plants

Mixed-integer nonlinear nonconvex programming problem

- $\approx$ 1000 continuous variables and 1000 constraints
- No more than 100-200 binary variables

How are these problems normally tackled?

Additional network elements

Elements that require the use of binary variables

- Different types of **control valves**
- **Operational ranges** of each compressor station
- Boil-off gas at regasification plants

Mixed-integer nonlinear nonconvex programming problem

- $\approx$ 1000 continuous variables and 1000 constraints
- No more than 100-200 binary variables

How are these problems normally tackled?

Two-step algorithms

Additional network elements

Elements that require the use of binary variables

- Different types of **control valves**
- **Operational ranges** of each compressor station
- Boil-off gas at regasification plants

Mixed-integer nonlinear nonconvex programming problem

- $\approx$ 1000 continuous variables and 1000 constraints
- No more than 100-200 binary variables

How are these problems normally tackled?

Two-step algorithms

- **Step 1.** Study a **simplified version** of the problem to fix all binary choices

Additional network elements

Elements that require the use of binary variables

- Different types of **control valves**
- **Operational ranges** of each compressor station
- Boil-off gas at regasification plants

Mixed-integer nonlinear nonconvex programming problem

- $\approx$ 1000 continuous variables and 1000 constraints
- No more than 100-200 binary variables

How are these problems normally tackled?

Two-step algorithms

- **Step 1.** Study a **simplified version** of the problem to fix all binary choices
- **Step 2.** Apply **SLP**, **SQP**,... to the resulting **continuous problem**

Our two-step approach for MINLP problems

Classic SLP

- We get a solution using **Classic SLP**

Our two-step approach for MINLP problems

Classic SLP

- **Step 2. Classic SLP.** Binary variables already **fixed**
  - We get a solution using **Classic SLP**

Our two-step approach for MINLP problems

Classic SLP

- **Step 1.**
- **Step 2. Classic SLP.** Binary variables already **fixed**
  - We get a solution using **Classic SLP**

Our two-step approach for MINLP problems

2SLP: **SLP-NTR** + **Classic SLP**

- **Step 1. SLP-NTR** (**N**o **T**rust **R**egion)
- **Step 2. Classic SLP.** Binary variables already **fixed**
  - We get a solution using **Classic SLP**

Our two-step approach for MINLP problems

2SLP: **SLP-NTR + Classic SLP**

- **Step 1. SLP-NTR** (**N**o **T**rust **R**egion)
  - The solution of this step is used to **fix** the binary variables
- **Step 2. Classic SLP.** Binary variables already **fixed**
  - We get a solution using **Classic SLP**

Our two-step approach for MINLP problems

2SLP: SLP-NTR + Classic SLP

- **Step 1. SLP-NTR** (**N**o **T**rust **R**egion)
  - The solution of this step is used to **fix** the binary variables
- **Step 2. Classic SLP.** Binary variables already **fixed**
  - We get a solution using **Classic SLP**

Step 1 runs on the full model. No simplification needed

SLP-NTR (No Trust Region)

Nonlinear programming problem: **NLP**

$$\begin{aligned} \textbf{minimize} \quad & f(\boldsymbol{x}) \\ \textbf{subject to} \quad & \\ \textbf{inequality contraints} \quad & g_i(\boldsymbol{x}) \leq 0, \quad i = 1, \cdots, m \\ \textbf{equality constrains} \quad & h_j(\boldsymbol{x}) = 0, \quad j = 1, \cdots, l \\ \textbf{linear constraints} \quad & \boldsymbol{x} \in X = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq b\} \end{aligned}$$

where f, g_i and h_j are **nonlinear** functions.

SLP-NTR (**N**o **T**rust **R**egion)

Classic SLP

SLP-NTR (No Trust Region)

Classic SLP

- At iteration k we have a candidate solution $\boldsymbol{x}^k$

SLP-NTR (No Trust Region)

Classic SLP

- At iteration k we have a candidate solution $\boldsymbol{x}^k$
- We solve the linearization of **NLP** about $\boldsymbol{x}^k$, LP($\boldsymbol{x}^k$):

SLP-NTR (No Trust Region)

Classic SLP

- At iteration k we have a candidate solution $\boldsymbol{x}^k$
- We solve the linearization of **NLP** about $\boldsymbol{x}^k$, LP($\boldsymbol{x}^k$):

$$\begin{array}{rll} \textbf{minimize} & \nabla f(\boldsymbol{x}^k)^t \boldsymbol{x} & \\ \textbf{subject to} & & \\ \textbf{inequality constraints} & g_i(\boldsymbol{x}^k) + \nabla g_i(\boldsymbol{x}^k)^t(\boldsymbol{x} - \boldsymbol{x}^k) \leq 0 & i = 1, \cdots, m \\ \textbf{equality constraints} & h_j(\boldsymbol{x}^k) + \nabla h_j(\boldsymbol{x}^k)^t(\boldsymbol{x} - \boldsymbol{x}^k) = 0 & j = 1, \cdots, l \\ \textbf{linear constraints} & \boldsymbol{x} \in X = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq b\} & \\ \textbf{trust region} & -d_k \leq \boldsymbol{x} - \boldsymbol{x}^k \leq d_k & \end{array}$$

SLP-NTR (No Trust Region)

Classic SLP

- At iteration k we have a candidate solution $\boldsymbol{x}^k$
- We solve the linearization of **NLP** about $\boldsymbol{x}^k$, LP($\boldsymbol{x}^k$):

$$
\begin{array}{rll}
\textbf{minimize} & \nabla f(\boldsymbol{x}^k)^t \boldsymbol{x} & \\
\textbf{subject to} & & \\
\textbf{inequality constraints} & g_i(\boldsymbol{x}^k) + \nabla g_i(\boldsymbol{x}^k)^t(\boldsymbol{x} - \boldsymbol{x}^k) \leq 0 & i = 1, \cdots, m \\
\textbf{equality constraints} & h_j(\boldsymbol{x}^k) + \nabla h_j(\boldsymbol{x}^k)^t(\boldsymbol{x} - \boldsymbol{x}^k) = 0 & j = 1, \cdots, l \\
\textbf{linear constraints} & \boldsymbol{x} \in X = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq b\} & \\
\textbf{trust region} & -d_k \leq \boldsymbol{x} - \boldsymbol{x}^k \leq d_k &
\end{array}
$$

SLP-NTR (No Trust Region)

Classic SLP

- At iteration k we have a candidate solution $\boldsymbol{x}^k$
- We solve the linearization of **NLP** about $\boldsymbol{x}^k$, LP($\boldsymbol{x}^k$):

$$
\begin{array}{rll}
\textbf{minimize} & \nabla f(\boldsymbol{x}^k)^t \boldsymbol{x} & \\
\textbf{subject to} & & \\
\textbf{inequality constraints} & g_i(\boldsymbol{x}^k) + \nabla g_i(\boldsymbol{x}^k)^t(\boldsymbol{x} - \boldsymbol{x}^k) \leq 0 & i = 1, \cdots, m \\
\textbf{equality constraints} & h_j(\boldsymbol{x}^k) + \nabla h_j(\boldsymbol{x}^k)^t(\boldsymbol{x} - \boldsymbol{x}^k) = 0 & j = 1, \cdots, l \\
\textbf{linear constraints} & \boldsymbol{x} \in X = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq b\} & \\
\textbf{trust region} & -d_k \leq \boldsymbol{x} - \boldsymbol{x}^k \leq d_k &
\end{array}
$$

Hard to accommodate binary variables with the trust region

SLP-NTR (No Trust Region)

SLP-NTR

- At iteration k we have a candidate solution $\boldsymbol{x}^k$
- We solve the linearization of **NLP** about $\boldsymbol{x}^k$, LP($\boldsymbol{x}^k$):

$$
\begin{array}{rll}
\textbf{minimize} & \nabla f(\boldsymbol{x}^k)^t \boldsymbol{x} & \\
\textbf{subject to} & & \\
\textbf{inequality constraints} & g_i(\boldsymbol{x}^k) + \nabla g_i(\boldsymbol{x}^k)^t(\boldsymbol{x} - \boldsymbol{x}^k) \leq 0 & i = 1, \cdots, m \\
\textbf{equality constraints} & h_j(\boldsymbol{x}^k) + \nabla h_j(\boldsymbol{x}^k)^t(\boldsymbol{x} - \boldsymbol{x}^k) = 0 & j = 1, \cdots, l \\
\textbf{linear constraints} & \boldsymbol{x} \in X = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq b\} & \\
\textbf{trust region} & -d_k \leq \boldsymbol{x} - \boldsymbol{x}^k \leq d_k &
\end{array}
$$

SLP-NTR (No Trust Region)

SLP-NTR

- At iteration k we have a candidate solution $\boldsymbol{x}^k$
- We solve the linearization of **NLP** about $\boldsymbol{x}^k$, LP($\boldsymbol{x}^k$):

$$
\begin{array}{rll}
\textbf{minimize} & \nabla f(\boldsymbol{x}^k)^t \boldsymbol{x} & \\
\textbf{subject to} & & \\
\textbf{inequality constraints} & g_i(\boldsymbol{x}^k) + \nabla g_i(\boldsymbol{x}^k)^t(\boldsymbol{x} - \boldsymbol{x}^k) \leq 0 & i = 1, \cdots, m \\
\textbf{equality constraints} & h_j(\boldsymbol{x}^k) + \nabla h_j(\boldsymbol{x}^k)^t(\boldsymbol{x} - \boldsymbol{x}^k) = 0 & j = 1, \cdots, l \\
\textbf{linear constraints} & \boldsymbol{x} \in X = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq b\} & \\
\textbf{trust region} & \cancel{-d_k \leq \boldsymbol{x} - \boldsymbol{x}^k \leq d_k} &
\end{array}
$$

- We **remove** the constraints that define the **trust region**

SLP-NTR (No Trust Region)

SLP-NTR

- At iteration k we have a candidate solution $\boldsymbol{x}^k$
- We solve the linearization of **NLP** about $\boldsymbol{x}^k$, LP($\boldsymbol{x}^k$):

minimize	$\nabla f(\boldsymbol{x}^k)^t \boldsymbol{x}$	
subject to		
inequality constraints	$g_i(\boldsymbol{x}^k) + \nabla g_i(\boldsymbol{x}^k)^t(\boldsymbol{x} - \boldsymbol{x}^k) \leq 0$	$i = 1, \cdots, m$
equality constraints	$h_j(\boldsymbol{x}^k) + \nabla h_j(\boldsymbol{x}^k)^t(\boldsymbol{x} - \boldsymbol{x}^k) = 0$	$j = 1, \cdots, l$
linear constraints	$\boldsymbol{x} \in X = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq b\}$	
trust region	~~$-d_k \leq \boldsymbol{x} - \boldsymbol{x}^k \leq d_k$~~	

- We **remove** the constraints that define the **trust region**

Straightforward inclusion of binary variables

SLP-NTR (No Trust Region)

SLP-NTR

- At iteration k we have a candidate solution $\boldsymbol{x}^k$
- We solve the linearization of **NLP** about $\boldsymbol{x}^k$, LP($\boldsymbol{x}^k$):

minimize $\nabla f(\boldsymbol{x}^k)^t \boldsymbol{x}$

subject to

inequality constraints $g_i(\boldsymbol{x}^k) + \nabla g_i(\boldsymbol{x}^k)^t(\boldsymbol{x} - \boldsymbol{x}^k) \leq 0 \quad i = 1, \cdots, m$

equality constraints $h_j(\boldsymbol{x}^k) + \nabla h_j(\boldsymbol{x}^k)^t(\boldsymbol{x} - \boldsymbol{x}^k) = 0 \quad j = 1, \cdots, l$

linear constraints $\boldsymbol{x} \in X = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq b\}$

trust region ~~$-d_k \leq \boldsymbol{x} - \boldsymbol{x}^k \leq d_k$~~

- We **remove** the constraints that define the **trust region**

Straightforward inclusion of binary variables

Theoretical justification for the removal of the trust region?

SLP-NTR vs classic SLP (in the continuous case, NLP problems)

SLP-NTR vs classic SLP (in the continuous case, NLP problems)

Classic SLP

- ++ Accumulation points of the sequence are **KKT** points of **NLP**
- ++ In practice it normally converges

SLP-NTR vs classic SLP (in the continuous case, NLP problems)

Classic SLP

++ Accumulation points of the sequence are **KKT** points of **NLP**

++ In practice it normally converges

−− **A number of parameters** have to be tuned

−− Hard to accommodate **binary** variables

SLP-NTR vs classic SLP (in the continuous case, NLP problems)

Classic SLP

++ Accumulation points of the sequence are **KKT** points of **NLP**

++ In practice it normally converges

−− **A number of parameters** have to be tuned

−− Hard to accommodate **binary** variables

SLP-NTR (**N**o **T**rust **R**egion)

SLP-NTR vs classic SLP (in the continuous case, NLP problems)

Classic SLP

++ Accumulation points of the sequence are **KKT** points of **NLP**

++ In practice it normally converges

−− **A number of parameters** have to be tuned

−− Hard to accommodate **binary** variables

SLP-NTR (**N**o **T**rust **R**egion)

++ If the sequence converges, the limit is a **KKT** point of **NLP**

SLP-NTR vs classic SLP (in the continuous case, NLP problems)

Classic SLP

- ++ Accumulation points of the sequence are **KKT** points of **NLP**
- ++ In practice it normally converges
- −− **A number of parameters** have to be tuned
- −− Hard to accommodate **binary** variables

SLP-NTR (**N**o **T**rust **R**egion)

- ++ If the sequence converges, the limit is a **KKT** point of **NLP**
- −− Other accumulation points may not be **KKT** points of **NLP**

SLP-NTR vs classic SLP (in the continuous case, NLP problems)

Classic SLP

++ Accumulation points of the sequence are **KKT** points of **NLP**

++ In practice it normally converges

−− **A number of parameters** have to be tuned

−− Hard to accommodate **binary** variables

SLP-NTR (**N**o **T**rust **R**egion)

++ If the sequence converges, the limit is a **KKT** point of **NLP**

−− Other accumulation points may not be **KKT** points of **NLP**

−− It cannot converge to interior points of the feasible set

SLP-NTR vs classic SLP (in the continuous case, NLP problems)

Classic SLP

++ Accumulation points of the sequence are **KKT** points of **NLP**

++ In practice it normally converges

−− **A number of parameters** have to be tuned

−− Hard to accommodate **binary** variables

SLP-NTR (**N**o **T**rust **R**egion)

++ If the sequence converges, the limit is a **KKT** point of **NLP**

−− Other accumulation points may not be **KKT** points of **NLP**

−− It cannot converge to interior points of the feasible set $(\min_{x\in[-1,1]} x^2)$

SLP-NTR vs classic SLP (in the continuous case, NLP problems)

Classic SLP

++ Accumulation points of the sequence are **KKT** points of **NLP**

++ In practice it normally converges

−− **A number of parameters** have to be tuned

−− Hard to accommodate **binary** variables

SLP-NTR (**N**o **T**rust **R**egion)

++ If the sequence converges, the limit is a **KKT** point of **NLP**

−− Other accumulation points may not be **KKT** points of **NLP**

−− It cannot converge to interior points of the feasible set $(\min_{x\in[-1,1]} x^2)$

(Not so critical, since we run 2SLP: SLP-NTR+CSLP)

SLP-NTR vs classic SLP (in the continuous case, NLP problems)

Classic SLP

++ Accumulation points of the sequence are **KKT** points of **NLP**

++ In practice it normally converges

−− **A number of parameters** have to be tuned

−− Hard to accommodate **binary** variables

SLP-NTR (**N**o **T**rust **R**egion)

++ If the sequence converges, the limit is a **KKT** point of **NLP**

−− Other accumulation points may not be **KKT** points of **NLP**

−− It cannot converge to interior points of the feasible set ($\min_{x \in [-1,1]} x^2$)
(Not so critical, since we run 2SLP: SLP-NTR+CSLP)

−− Less stable in terms of convergence (*e.g.*, **cycling**)

SLP-NTR vs classic SLP (in the continuous case, NLP problems)

Classic SLP

++ Accumulation points of the sequence are **KKT** points of **NLP**

++ In practice it normally converges

−− **A number of parameters** have to be tuned

−− Hard to accommodate **binary** variables

SLP-NTR (**N**o **T**rust **R**egion)

++ If the sequence converges, the limit is a **KKT** point of **NLP**

−− Other accumulation points may not be **KKT** points of **NLP**

−− It cannot converge to interior points of the feasible set ($\min_{x\in[-1,1]} x^2$)
(Not so critical, since we run 2SLP: SLP-NTR+CSLP)

−− Less stable in terms of convergence (*e.g.*, **cycling**)

++ If two consecutive points of $\{x^k\}$
are sufficiently close $\longrightarrow$ almost KKT of **NLP**

SLP-NTR vs classic SLP (in the continuous case, NLP problems)

Classic SLP

++ Accumulation points of the sequence are **KKT** points of **NLP**

++ In practice it normally converges

−− **A number of parameters** have to be tuned

−− Hard to accommodate **binary** variables

SLP-NTR (**N**o **T**rust **R**egion)

++ If the sequence converges, the limit is a **KKT** point of **NLP**

−− Other accumulation points may not be **KKT** points of **NLP**

−− It cannot converge to interior points of the feasible set ($\min_{x\in[-1,1]} x^2$)
(Not so critical, since we run 2SLP: SLP-NTR+CSLP)

−− Less stable in terms of convergence (*e.g.*, **cycling**)

++ If two consecutive points of $\{x^k\}$
are sufficiently close $\longrightarrow$ almost KKT of **NLP**

SLP-NTR vs classic SLP (in the continuous case, NLP problems)

Classic SLP

++ Accumulation points of the sequence are **KKT** points of **NLP**

++ In practice it normally converges

−− **A number of parameters** have to be tuned

−− Hard to accommodate **binary** variables

SLP-NTR (**N**o **T**rust **R**egion)

++ If the sequence converges, the limit is a **KKT** point of **NLP**

−− Other accumulation points may not be **KKT** points of **NLP**

−− It cannot converge to interior points of the feasible set $(\min_{x \in [-1,1]} x^2)$
(Not so critical, since we run 2SLP: SLP-NTR+CSLP)

−− Less stable in terms of convergence (*e.g.*, **cycling**)

++ If two consecutive points of $\{\boldsymbol{x}^k\}$
are sufficiently close $\longrightarrow$ almost KKT of **NLP**

++ Very easy to implement. No parameters to be tuned

SLP-NTR vs classic SLP (in the continuous case, NLP problems)

Classic SLP

++ Accumulation points of the sequence are **KKT** points of **NLP**

++ In practice it normally converges

−− **A number of parameters** have to be tuned

−− Hard to accommodate **binary** variables

SLP-NTR (**N**o **T**rust **R**egion)

++ If the sequence converges, the limit is a **KKT** point of **NLP**

−− Other accumulation points may not be **KKT** points of **NLP**

−− It cannot converge to interior points of the feasible set $(\min_{x\in[-1,1]} x^2)$
(Not so critical, since we run 2SLP: SLP-NTR+CSLP)

−− Less stable in terms of convergence (*e.g.*, **cycling**)

++ If two consecutive points of $\{x^k\}$
are sufficiently close $\longrightarrow$ almost KKT of **NLP**

++ Very easy to implement. No parameters to be tuned

++ It is straightforward to incorporate **binary** variables

SLP-NTR vs classic SLP (in the continuous case, NLP problems)

Classic SLP

++ Accumulation points of the sequence are **KKT** points of **NLP**

++ In practice it normally converges

−− **A number of parameters** have to be tuned

−− Hard to accommodate **binary** variables

SLP-NTR (No Trust Region)

++ If the sequence converges, the limit is a **KKT** point of **NLP**

−− Other accumulation points may not be **KKT** points of **NLP**

−− It cannot converge to interior points of the feasible set ($\min_{x\in[-1,1]} x^2$)
(Not so critical, since we run 2SLP: SLP-NTR+CSLP)

−− Less stable in terms of convergence (*e.g.*, **cycling**)

++ If two consecutive points of $\{\boldsymbol{x}^k\}$
are sufficiently close $\longrightarrow$ almost KKT of **NLP**

++ Very easy to implement. No parameters to be tuned

++ It is straightforward to incorporate **binary** variables

++ SLP-NTR competitive with classic SLP for **gas network problems** and **multicommodity flow problems**

SLP-NTR vs classic SLP (in the continuous case, NLP problems)

Classic SLP

++ Accumulation points of the sequence are **KKT** points of **NLP**

++ In practice it normally converges

−− **A number of parameters** have to be tuned

−− Hard to accommodate **binary** variables

SLP-NTR (**N**o **T**rust **R**egion)

++ If the sequence converges, the limit is a **KKT** point of **NLP**

−− Other accumulation points may not be **KKT** points of **NLP**

−− It cannot converge to interior points of the feasible set ($\min_{x\in[-1,1]} x^2$)
(Not so critical, since we run 2SLP: SLP-NTR+CSLP)

−− Less stable in terms of convergence (*e.g.*, **cycling**)

++ If two consecutive points of $\{\boldsymbol{x}^k\}$ are sufficiently close $\longrightarrow$ almost KKT of **NLP**

++ Very easy to implement. No parameters to be tuned

++ It is straightforward to incorporate **binary** variables

++ SLP-NTR competitive with classic SLP for **gas network problems** and **multicommodity flow problems**

Summary (algorithm for MINLP problems)

2SLP: SLP-NTR + Classic SLP

- **Step 1. SLP-NTR** (**N**o **T**rust **R**egion)
- **Step 2. Classic SLP**

Summary (algorithm for MINLP problems)

2SLP: SLP-NTR + Classic SLP

- **Step 1. SLP-NTR** (**N**o **T**rust **R**egion)
- **Step 2. Classic SLP**

Features of our two-step approach

Summary (algorithm for MINLP problems)

2SLP: **SLP-NTR + Classic SLP**

- **Step 1. SLP-NTR** (**N**o **T**rust **R**egion)
- **Step 2. Classic SLP**

Features of our two-step approach

- Easy to implement

Summary (algorithm for MINLP problems)

2SLP: **SLP-NTR + Classic SLP**

- **Step 1. SLP-NTR** (**N**o **T**rust **R**egion)
- **Step 2. Classic SLP**

Features of our two-step approach

- Easy to implement
- **Step 1** runs on the **full model**. No simplification needed

Summary (algorithm for MINLP problems)

2SLP: SLP-NTR + Classic SLP

- **Step 1. SLP-NTR** (**N**o **T**rust **R**egion)
- **Step 2. Classic SLP**

Features of our two-step approach

- Easy to implement
- **Step 1** runs on the **full model**. No simplification needed
- **Step 2** "guarantees" convergence

Summary (algorithm for MINLP problems)

2SLP: **SLP-NTR + Classic SLP**

- **Step 1. SLP-NTR** (**N**o **T**rust **R**egion)
- **Step 2. Classic SLP**

Features of our two-step approach

- Easy to implement
- **Step 1** runs on the **full model**. No simplification needed
- **Step 2** "guarantees" convergence
- Good practical behavior (< 5 minutes running time on Spanish network)
  - **Significant cost reduction** with respect to operation schemes reported by the Transmission System Operator (whose software does not optimize)

Summary (algorithm for MINLP problems)

2SLP: SLP-NTR + Classic SLP

- **Step 1. SLP-NTR** (**N**o **T**rust **R**egion)
- **Step 2. Classic SLP**

Features of our two-step approach

- Easy to implement
- **Step 1** runs on the **full model**. No simplification needed
- **Step 2** “guarantees” convergence
- Good practical behavior (< 5 minutes running time on Spanish network)
  - **Significant cost reduction** with respect to operation schemes reported by the Transmission System Operator (whose software does not optimize)
- Limitation: **No bounds/gap** to optimality

Our contribution

Our contribution

NLP problems

Theoretical foundation for the **SLP-NTR algorithm**

Our contribution

NLP problems

Theoretical foundation for the **SLP-NTR algorithm**

MINLP problems

Heuristic approach based on the **SLP-NTR algorithm**

Our contribution

NLP problems

Theoretical foundation for the **SLP-NTR algorithm**

MINLP problems

Heuristic approach based on the **SLP-NTR algorithm**

- Nothing deep, but we have not seen it elsewhere

Our contribution

NLP problems

Theoretical foundation for the **SLP-NTR algorithm**

MINLP problems

Heuristic approach based on the **SLP-NTR algorithm**

- Nothing deep, but we have not seen it elsewhere
- Good performance in real size problems

Our contribution

NLP problems

Theoretical foundation for the **SLP-NTR algorithm**

MINLP problems

Heuristic approach based on the **SLP-NTR algorithm**

- Nothing deep, but we have not seen it elsewhere
- Good performance in real size problems

MINLP stochastic problems

Our contribution

NLP problems

Theoretical foundation for the **SLP-NTR algorithm**

MINLP problems

Heuristic approach based on the **SLP-NTR algorithm**

- Nothing deep, but we have not seen it elsewhere
- Good performance in real size problems

MINLP stochastic problems

- Long-term **infrastructure planning under uncertainty** (prices and demands)

Our contribution

NLP problems

Theoretical foundation for the **SLP-NTR algorithm**

MINLP problems

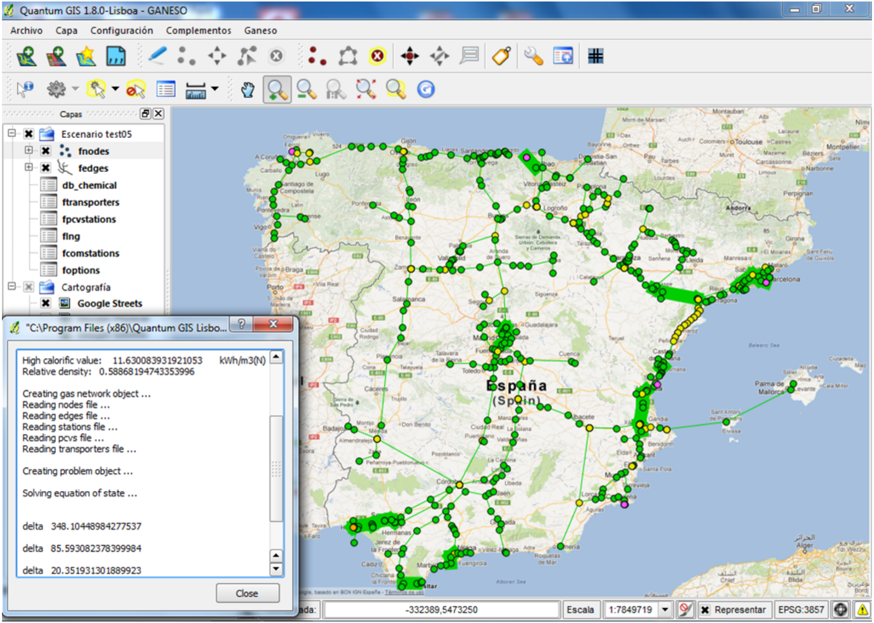
Heuristic approach based on the **SLP-NTR algorithm**

- Nothing deep, but we have not seen it elsewhere
- Good performance in real size problems

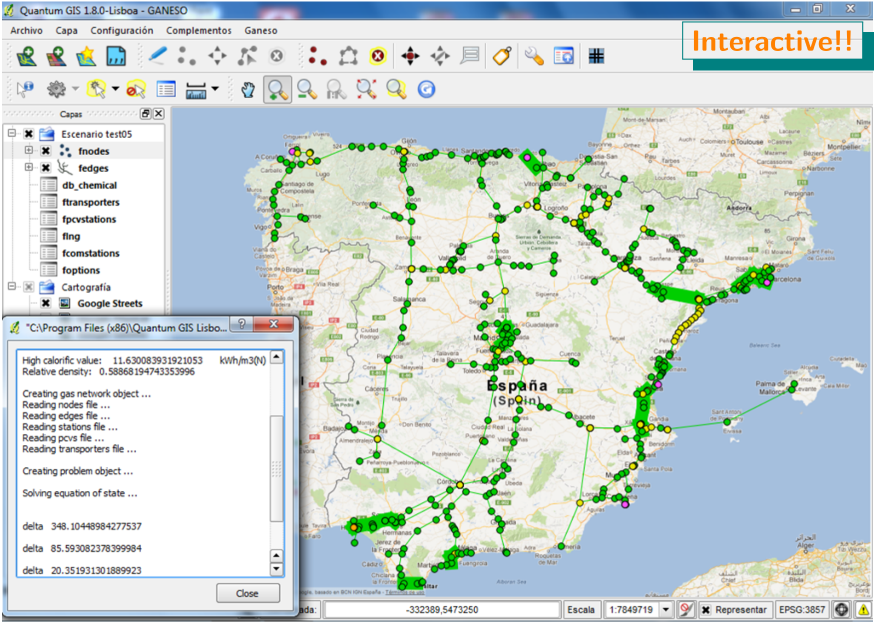
MINLP stochastic problems

- Long-term **infrastructure planning under uncertainty** (prices and demands)
- Implementation of a **lagrangian decomposition algorithm** (progressive hedging) that uses **SLP-NTR algorithm** to solve the MINLP subproblems

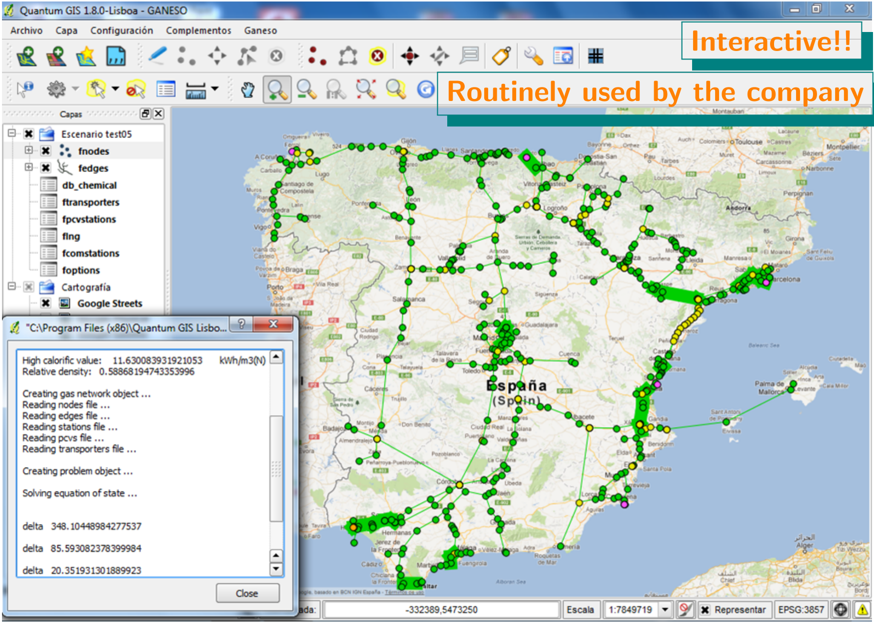
GANESO^{TM} user interface



GANESO$^{\mathrm{TM}}$ user interface



GANESO™ user interface



Interactive!!

Routinely used by the company

Numerical results

1. Comparisons on the Spanish gas transmission network
2. Comparisons on related gas transmission problems
3. Comparisons on multicommodity flow problems

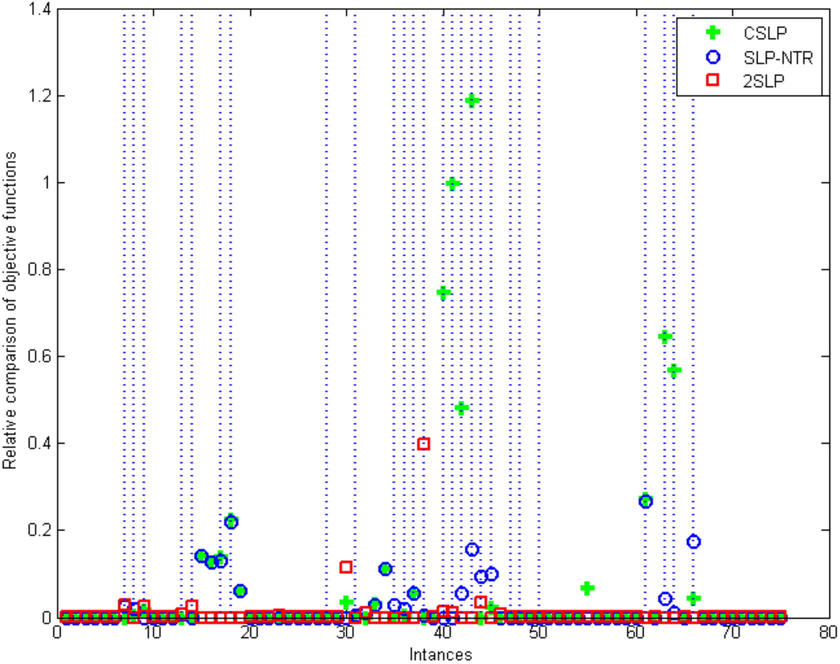
Numerical results

1. Comparisons on the Spanish gas transmission network
2. Comparisons on related gas transmission problems
3. Comparisons on multicommodity flow problems

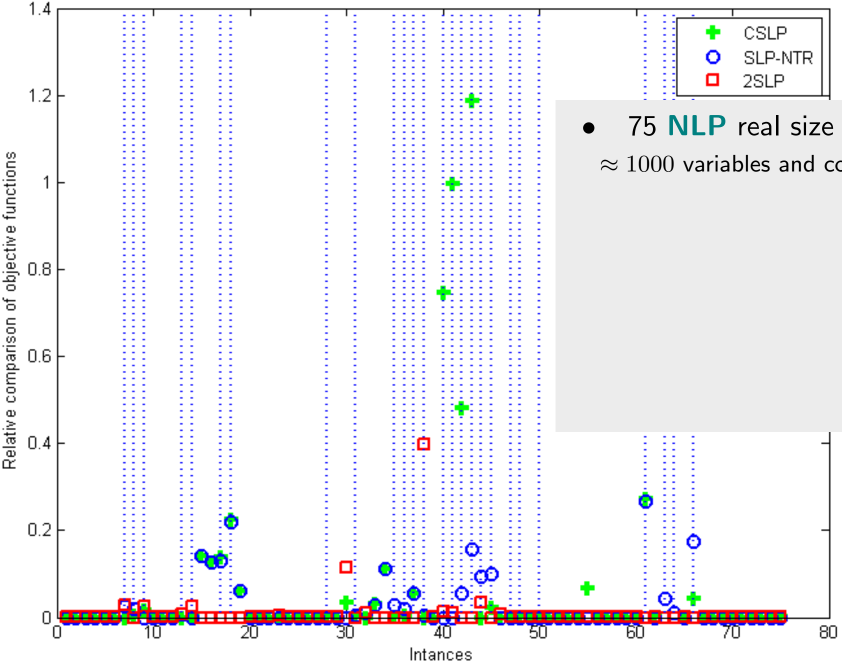
Work in progress

Tests on the Spanish Gas Transmission Network (NLP)

Tests on the Spanish Gas Transmission Network (NLP)



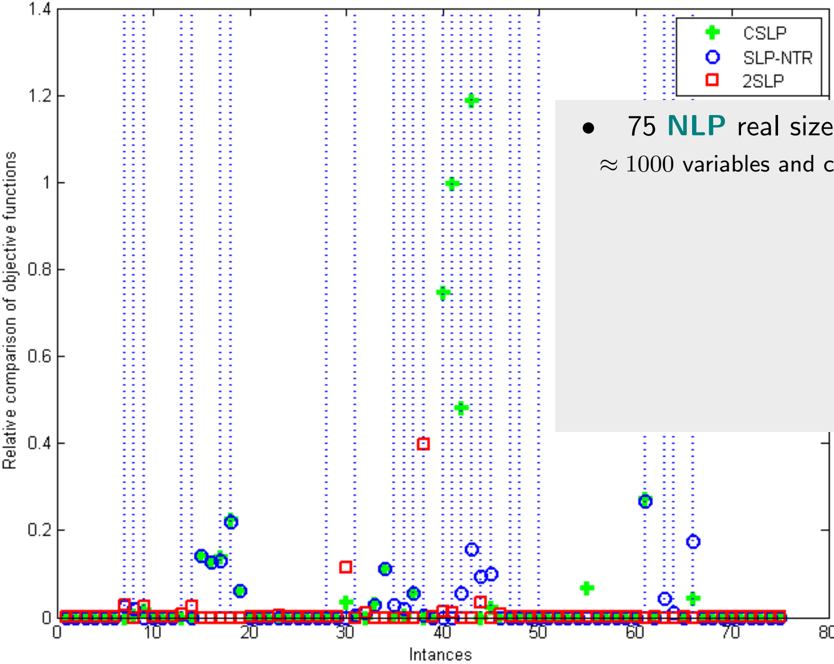
Tests on the Spanish Gas Transmission Network (NLP)



- 75 **NLP** real size instances

  ≈ 1000 variables and constraints

Tests on the Spanish Gas Transmission Network (NLP)

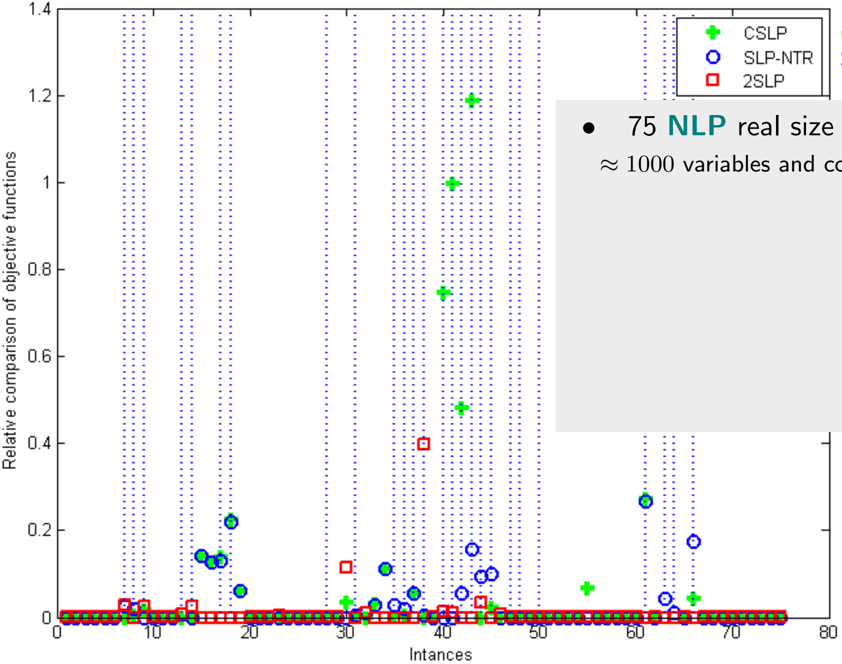


CSLP

- 75 **NLP** real size instances

  ≈ 1000 variables and constraints

Tests on the Spanish Gas Transmission Network (NLP)

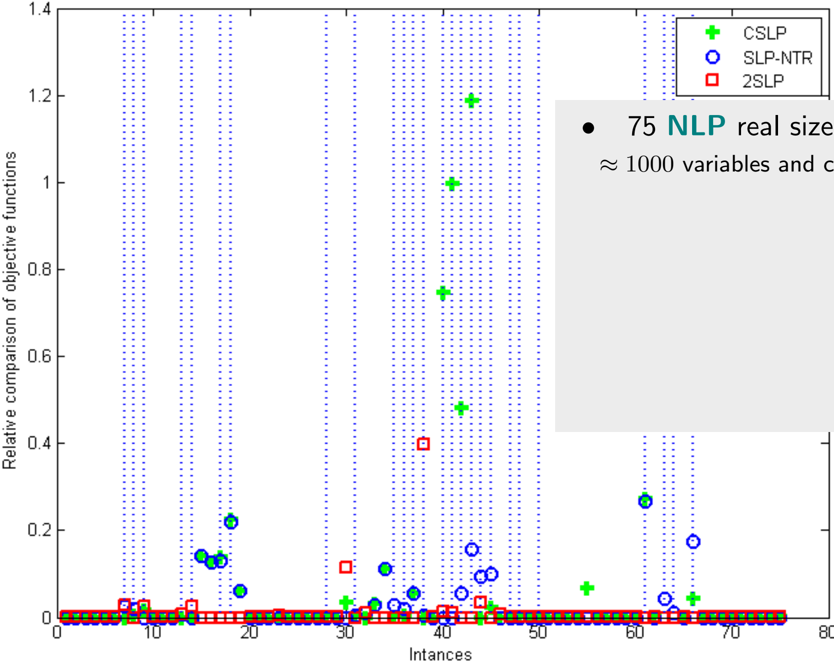


CSLP
SLP-NTR

- 75 **NLP** real size instances

  ≈ 1000 variables and constraints

Tests on the Spanish Gas Transmission Network (NLP)

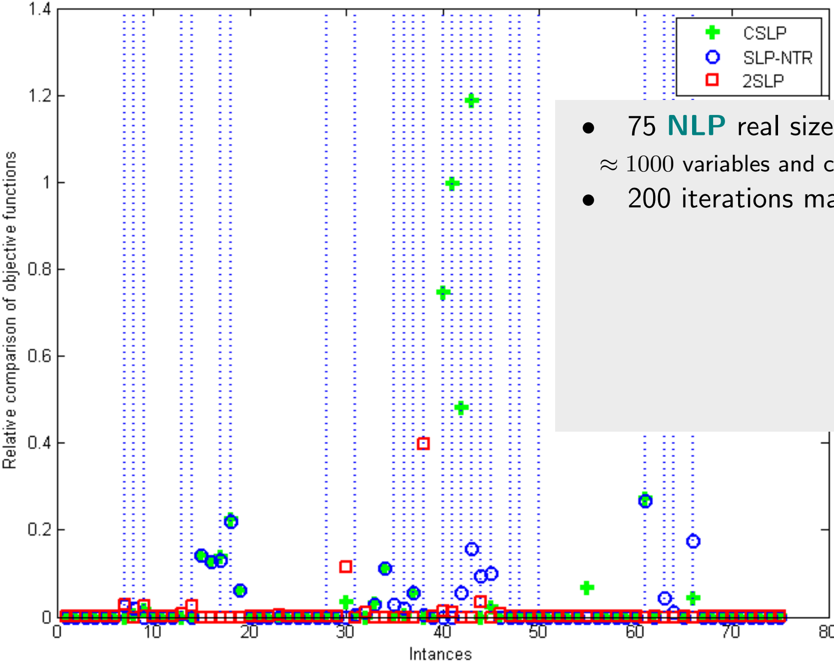


CSLP
SLP-NTR
2SLP

- 75 **NLP** real size instances

  ≈ 1000 variables and constraints

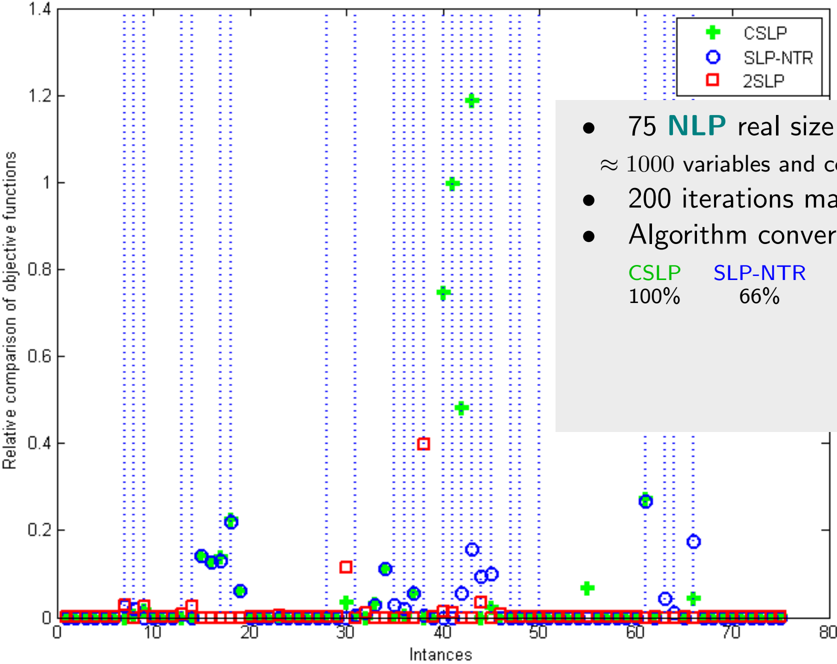
Tests on the Spanish Gas Transmission Network (NLP)



CSLP
SLP-NTR
2SLP

- 75 **NLP** real size instances
  ≈ 1000 variables and constraints
- 200 iterations maximum

Tests on the Spanish Gas Transmission Network (NLP)

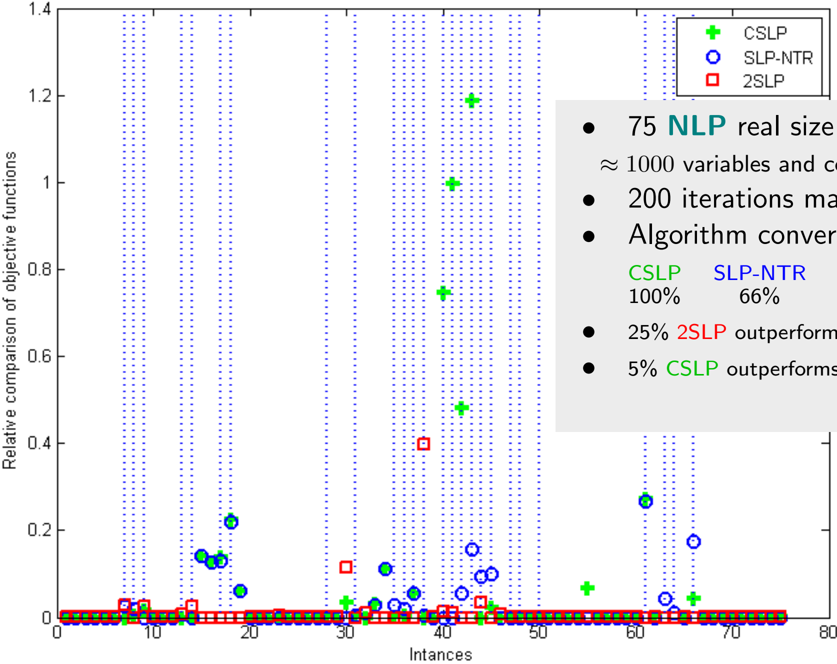


CSLP
SLP-NTR
2SLP

- 75 **NLP** real size instances
  ≈ 1000 variables and constraints
- 200 iterations maximum
- Algorithm convergence:

CSLP	SLP-NTR	2SLP
100%	66%	100%

Tests on the Spanish Gas Transmission Network (NLP)



CSLP
SLP-NTR
2SLP

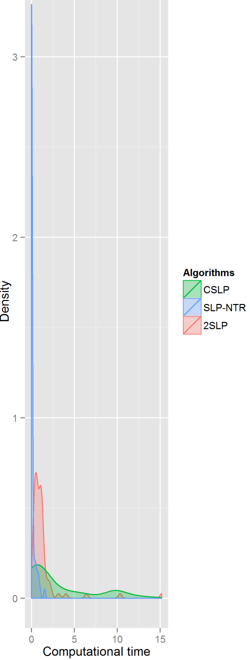
- 75 **NLP** real size instances

  ≈ 1000 variables and constraints
- 200 iterations maximum
- Algorithm convergence:

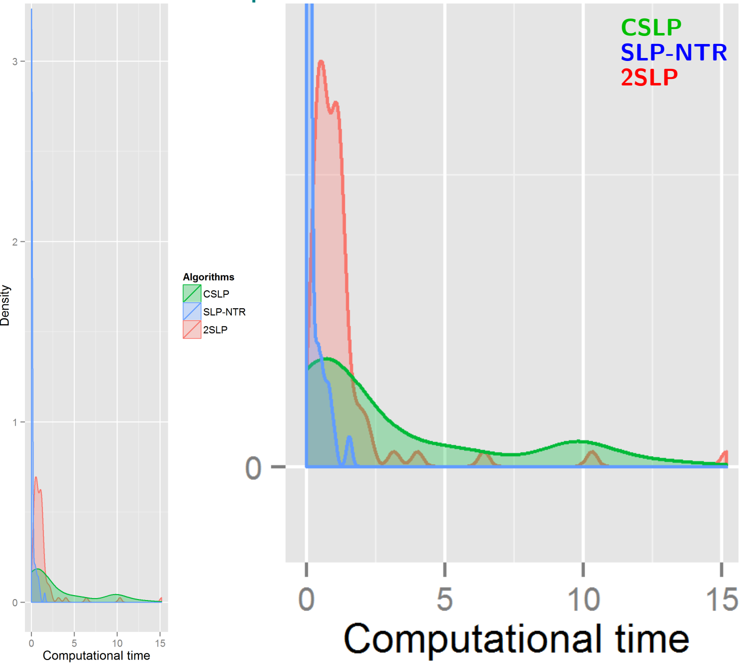
CSLP	SLP-NTR	2SLP
100%	66%	100%

- 25% 2SLP outperforms CSLP
- 5% CSLP outperforms 2SLP

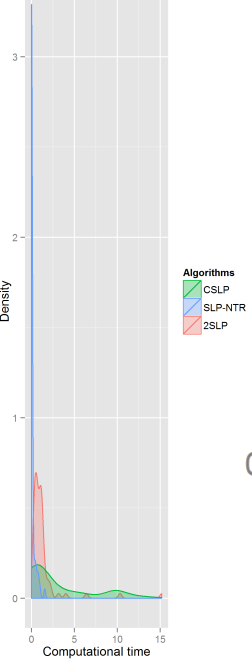
Tests on the Spanish Gas Transmission Network (NLP)

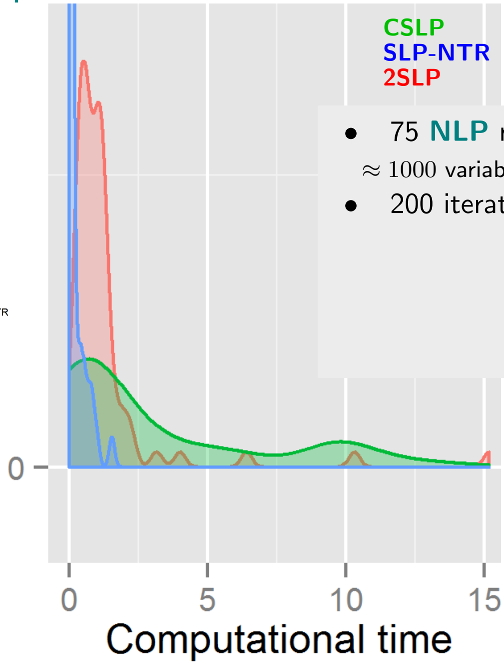


Tests on the Spanish Gas Transmission Network (NLP)



Tests on the Spanish Gas Transmission Network (NLP)

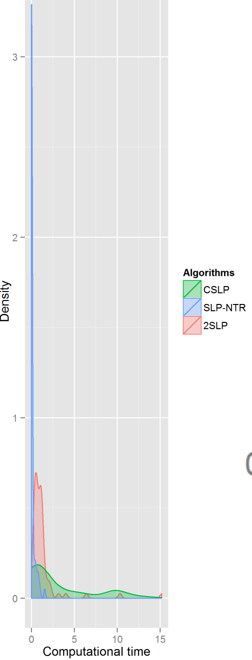


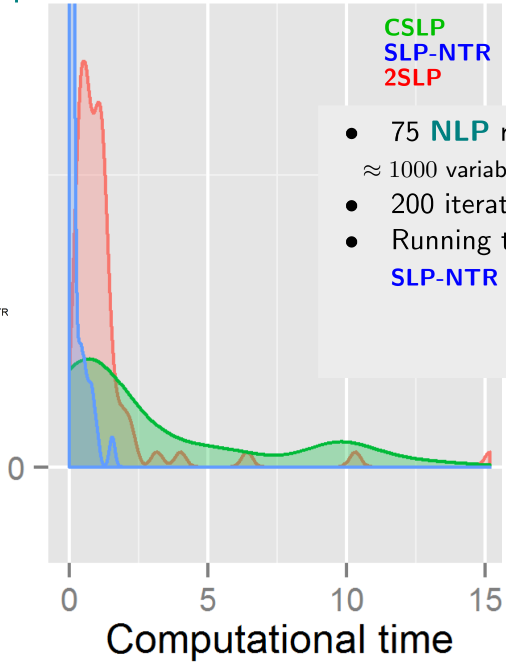


- 75 **NLP** real size instances

  ≈ 1000 variables and constraints
- 200 iterations maximum

Tests on the Spanish Gas Transmission Network (NLP)



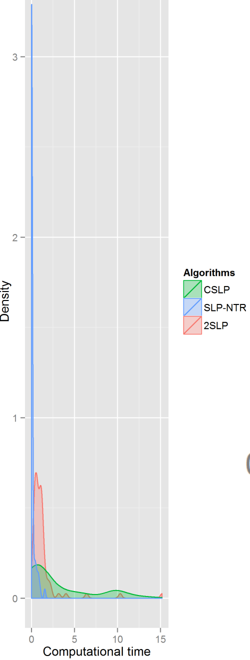


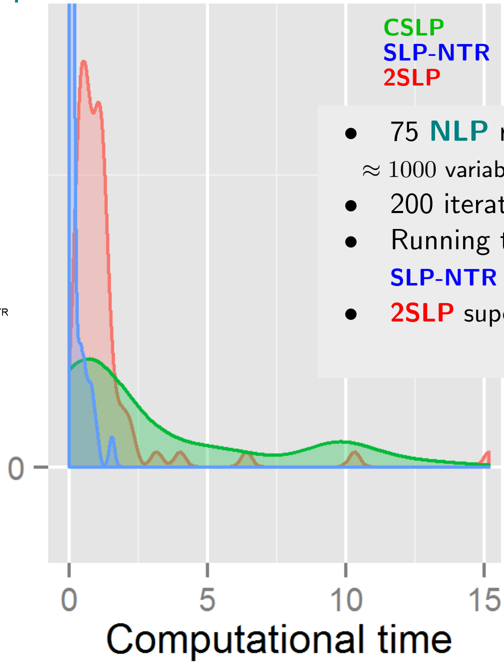
- 75 **NLP** real size instances

  ≈ 1000 variables and constraints
- 200 iterations maximum
- Running times:

  SLP-NTR $\ll$ **2SLP** $\ll$ **CSLP**

Tests on the Spanish Gas Transmission Network (NLP)





- 75 **NLP** real size instances
  ≈ 1000 variables and constraints
- 200 iterations maximum
- Running times:
  SLP-NTR $\ll$ **2SLP** $\ll$ **CSLP**
- **2SLP** superior performance

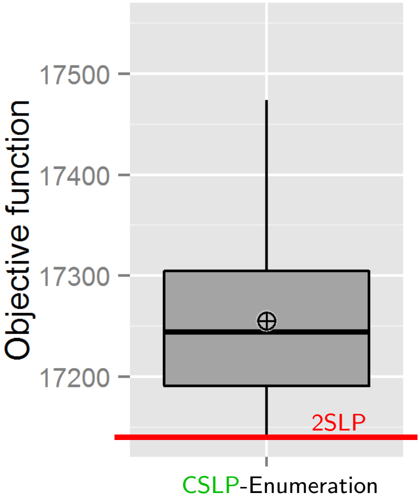
Tests on the Spanish Gas Transmission Network (MINLP)

Tests on the Spanish Gas Transmission Network (MINLP)

- Real size instance with 10 binary variables
- 2SLP vs CSLP-Enumeration

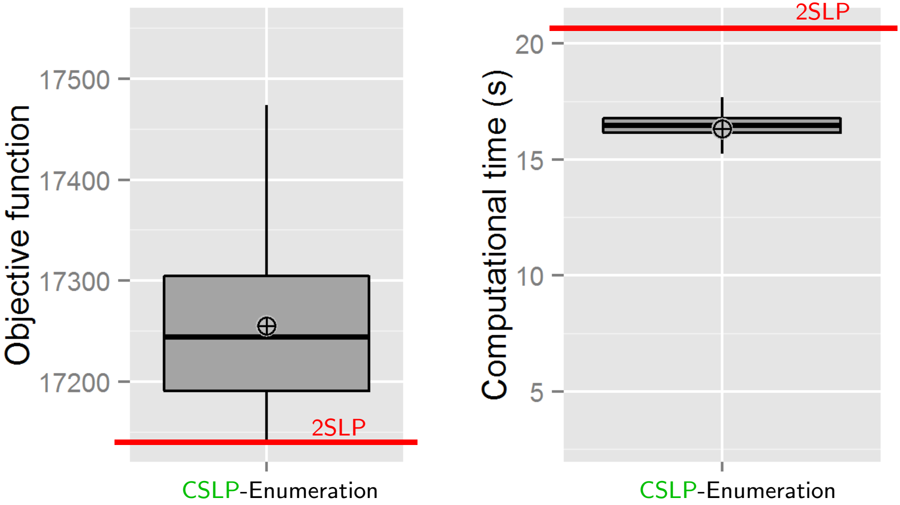
Tests on the Spanish Gas Transmission Network (MINLP)

- Real size instance with 10 binary variables
- 2SLP vs CSLP-Enumeration



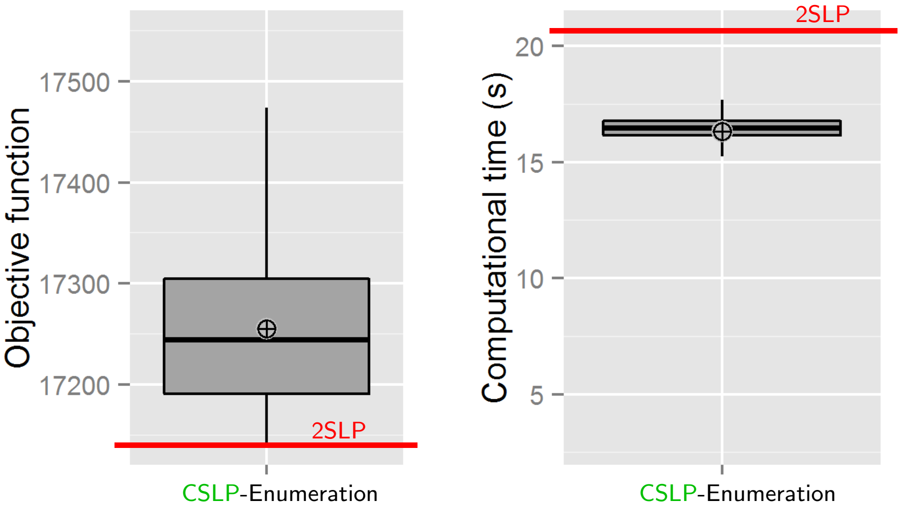
Tests on the Spanish Gas Transmission Network (MINLP)

- Real size instance with 10 binary variables
- 2SLP vs CSLP-Enumeration



Tests on the Spanish Gas Transmission Network (MINLP)

- Real size instance with 10 binary variables
- 2SLP vs CSLP-Enumeration



- **Next task.** Designing a full set of test instances

Tests on the Belgian Gas Transmission Network

(de Wolfe and Smeers, 2000)

- Slightly different model of the gas transmission problem
- Small example: ≈ 50 variables and constraints

Tests on the Belgian Gas Transmission Network
(de Wolfe and Smeers, 2000)

- Slightly different model of the gas transmission problem
- Small example: ≈ 50 variables and constraints

Tests on the Belgian Gas Transmission Network
(de Wolfe and Smeers, 2000)

- Slightly different model of the gas transmission problem
- Small example: ≈ 50 variables and constraints

NLP formulation of the problem

NLP problem	**CSLP**	**SLP-NTR**	**2SLP**	BARON	Knitro
Objective function	91.0562	91.0562	91.0562	91.0562	91.0562
Computational time	0.3654	0.3570	0.3570	0.0733	0.0093

Tests on the Belgian Gas Transmission Network

(de Wolfe and Smeers, 2000)

- Slightly different model of the gas transmission problem
- Small example: ≈ 50 variables and constraints

NLP formulation of the problem

NLP problem	**CSLP**	**SLP-NTR**	**2SLP**	BARON	Knitro
Objective function	91.0562	91.0562	91.0562	91.0562	91.0562
Computational time	0.3654	0.3570	0.3570	0.0733	0.0093

MINLP formulation of the problem

- $|q_{ij}|q_{ij}$. The absolute values in the constraints are modeled using binary variables that account for the sign of q_{ij}

Tests on the Belgian Gas Transmission Network

(de Wolfe and Smeers, 2000)

- Slightly different model of the gas transmission problem
- Small example: $\approx$ 50 variables and constraints

NLP formulation of the problem

NLP problem	**CSLP**	**SLP-NTR**	**2SLP**	BARON	Knitro
Objective function	91.0562	91.0562	91.0562	91.0562	91.0562
Computational time	0.3654	0.3570	0.3570	0.0733	0.0093

MINLP formulation of the problem

- $|q_{ij}|q_{ij}$. The absolute values in the constraints are modeled using binary variables that account for the sign of q_{ij}
- ≈ 25 binary variables and 50 additional constraints

Tests on the Belgian Gas Transmission Network

(de Wolfe and Smeers, 2000)

- Slightly different model of the gas transmission problem
- Small example: ≈ 50 variables and constraints

NLP formulation of the problem

NLP problem	**CSLP**	**SLP-NTR**	**2SLP**	BARON	Knitro
Objective function	91.0562	91.0562	91.0562	91.0562	91.0562
Computational time	0.3654	0.3570	0.3570	0.0733	0.0093

MINLP formulation of the problem

- $|q_{ij}|q_{ij}$. The absolute values in the constraints are modeled using binary variables that account for the sign of q_{ij}
- ≈ 25 binary variables and 50 additional constraints

NLP problem	**2SLP**	BARON	Knitro
Objective function	91.0562	91.0562	94.8715 (infeasible)
Computational time	0.6497	231.2602	0.0983

Tests on the Belgian Gas Transmission Network

(de Wolfe and Smeers, 2000)

- Slightly different model of the gas transmission problem
- Small example: $\approx$ 50 variables and constraints

NLP formulation of the problem

NLP problem	**CSLP**	**SLP-NTR**	**2SLP**	BARON	Knitro
Objective function	91.0562	91.0562	91.0562	91.0562	91.0562
Computational time	0.3654	0.3570	0.3570	0.0733	0.0093

MINLP formulation of the problem

- $|q_{ij}|q_{ij}$. The absolute values in the constraints are modeled using binary variables that account for the sign of q_{ij}
- ≈ 25 binary variables and 50 additional constraints

NLP problem	**2SLP**	BARON	Knitro
Objective function	91.0562	91.0562	94.8715 (infeasible)
Computational time	0.6497	231.2602	0.0983

Tests on the Belgian Gas Transmission Network
(de Wolfe and Smeers, 2000)

- Slightly different model of the gas transmission problem
- Small example: $\approx$ 50 variables and constraints

NLP formulation of the problem

NLP problem	**CSLP**	**SLP-NTR**	**2SLP**	BARON	Knitro
Objective function	91.0562	91.0562	91.0562	91.0562	91.0562
Computational time	0.3654	0.3570	0.3570	0.0733	0.0093

MINLP formulation of the problem

- $|q_{ij}|q_{ij}$. The absolute values in the constraints are modeled using binary variables that account for the sign of q_{ij}
- ≈ 25 binary variables and 50 additional constraints

NLP problem	**2SLP**	BARON	Knitro
Objective function	91.0562	91.0562	94.8715 (infeasible)
Computational time	0.6497	231.2602	0.0983

Tests on the Belgian Gas Transmission Network
(de Wolfe and Smeers, 2000)

- Slightly different model of the gas transmission problem
- Small example: ≈ 50 variables and constraints

NLP formulation of the problem

NLP problem	CSLP	SLP-NTR	2SLP	BARON	Knitro
Objective function	91.0562	91.0562	91.0562	91.0562	91.0562
Computational time	0.3654	0.3570	0.3570	0.0733	0.0093

MINLP formulation of the problem

- $|q_{ij}|q_{ij}$. The absolute values in the constraints are modeled using binary variables that account for the sign of q_{ij}
- ≈ 25 binary variables and 50 additional constraints

NLP problem	2SLP	BARON	Knitro
Objective function	91.0562	91.0562	94.8715 (infeasible)
Computational time	0.6497	231.2602	0.0983

- **Next task.** Designing a full set of test instances

Tests on multicommodity flow problems (NLP)

- Linear constraints and **nonlinear** objective function

Tests on multicommodity flow problems (NLP)

- Linear constraints and **nonlinear** objective function (feasibility ✓)

Tests on multicommodity flow problems (NLP)

- Linear constraints and **nonlinear** objective function (feasibility ✓)
- Benchmark test sets available (Babonneau et al. 2004)

Tests on multicommodity flow problems (NLP)

- Linear constraints and **nonlinear** objective function (feasibility ✓)
- Benchmark test sets available (Babonneau et al. 2004)

Problem	$\|N\|$	$\|E\|$	$\|T\|$	Constr.	Variab.	z_{opt}	Relative error		
Planar problems							**CSLP**	**SLP-NTR**	**2SLP**
P30	30	150	92	2760	13800	4.445×10^7	0.0074	0.0085	0.0074
P50	50	250	267	13350	66750	1.212×10^8	0.0202	0.0212	0.0202
P80	80	440	543	43440	238920	1.819×10^8	0.0174	0.0188	0.0174
P100	100	532	1085	108500	577220	2.291×10^8	0.0212	0.0219	0.0212
Grid problems									
G1	25	80	50	1250	4000	8.336×10^5	0.0003	0.0054	0.0004
G2	25	80	100	2500	8000	1.727×10^6	0.0006	0.0089	0.0005
G3	100	360	50	5000	18000	1.532×10^6	0.0000	0.0065	0.0002
G4	100	360	100	10000	36000	3.055×10^6	0.0000	0.0066	0.0000
G5	225	840	100	22500	84000	5.079×10^6	0.0000	0.0069	0.0000
G6	225	840	200	45000	168000	1.051×10^7	0.0001	0.0108	0.0002
G7	400	1520	400	160000	608000	2.607×10^7	0.0000	0.0031	0.0000
Telecommunication-like problems									
N22	14	22	23	322	506	1.871×10^3	0.0131	0.0131	0.0131
N148	58	148	122	7076	18056	1.402×10^5	0.0000	0.0002	0.0000
Transportation problems									
S-F	24	76	528	12672	40128	3.202×10^5	0.0050	0.0051	0.0050

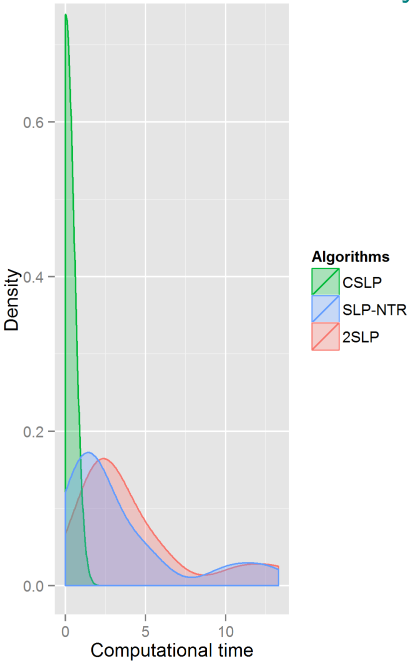
Tests on multicommodity flow problems (NLP)

- Linear constraints and **nonlinear** objective function (feasibility ✓)
- Benchmark test sets available (Babonneau et al. 2004)

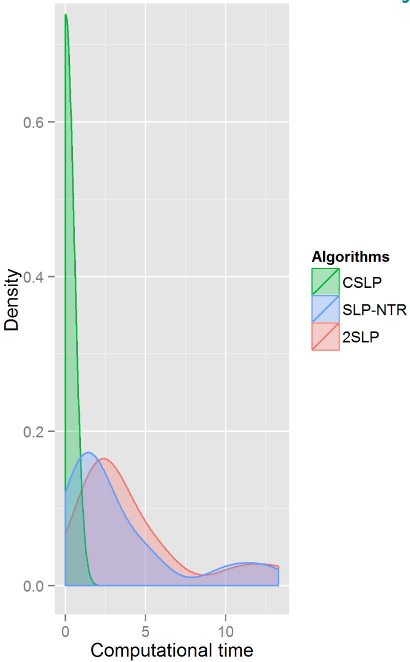
Problem	$\lvert N \rvert$	$\lvert E \rvert$	$\lvert T \rvert$	Constr.	Variab.	z_{opt}	Relative error		
Planar problems							CSLP	SLP-NTR	2SLP
P30	30	150	92	2760	13800	4.445×10^7	0.0074	0.0085	0.0074
P50	50	250	267	13350	66750	1.212×10^8	0.0202	0.0212	0.0202
P80	80	440	543	43440	238920	1.819×10^8	0.0174	0.0188	0.0174
P100	100	532	1085	108500	577220	2.291×10^8	0.0212	0.0219	0.0212
Grid problems									
G1	25	80	50	1250	4000	8.336×10^5	0.0003	0.0054	0.0004
G2	25	80	100	2500	8000	1.727×10^6	0.0006	0.0089	0.0005
G3	100	360	50	5000	18000	1.532×10^6	0.0000	0.0065	0.0002
G4	100	360	100	10000	36000	3.055×10^6	0.0000	0.0066	0.0000
G5	225	840	100	22500	84000	5.079×10^6	0.0000	0.0069	0.0000
G6	225	840	200	45000	168000	1.051×10^7	0.0001	0.0108	0.0002
G7	400	1520	400	160000	608000	2.607×10^7	0.0000	0.0031	0.0000
Telecommunication-like problems									
N22	14	22	23	322	506	1.871×10^3	0.0131	0.0131	0.0131
N148	58	148	122	7076	18056	1.402×10^5	0.0000	0.0002	0.0000
Transportation problems									
S-F	24	76	528	12672	40128	3.202×10^5	0.0050	0.0051	0.0050

- All approaches very competitive in terms of objective function

Tests on multicommodity flow problems (NLP)

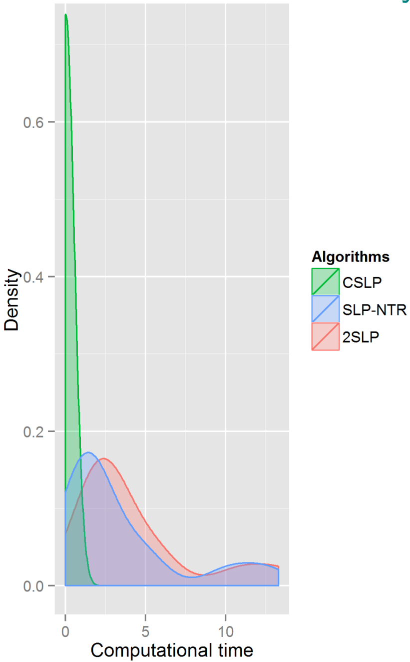


Tests on multicommodity flow problems (NLP)



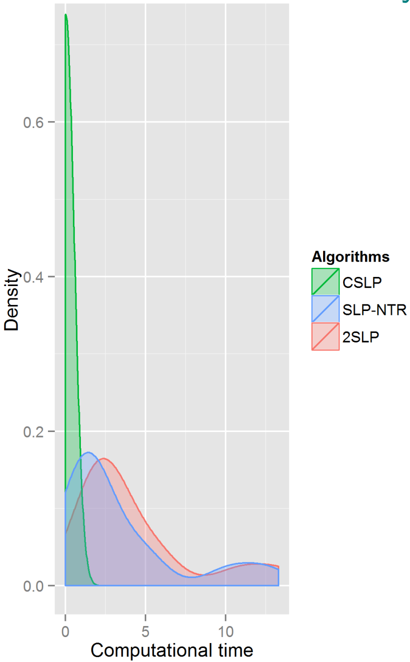
- Now **CSLP** is the fastest one

Tests on multicommodity flow problems (NLP)



- Now **CSLP** is the fastest one Why??

Tests on multicommodity flow problems (NLP)



- Now **CSLP** is the fastest one   Why??
- Apparently, the trust region helps to solve faster **very large** linearized **subproblems**

A two-step sequential linear programming algorithm for MINLP problems:

An application to gas transmission networks

Julio González-Díaz
Ángel M. González-Rueda
María P. Fernández de Córdoba

University of Santiago de Compostela
Technological Institute for Industrial Mathematics (ITMATI)

....

RSME2017
2ARA602A

February 3rd, 2017

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

itmati

reganosa