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
February 3rd, 2017
A two-step sequential linear programming algorithm for MINLP problems:
An application to gas transmission networks

1 Optimization in Gas Transmission Networks

2 (A twist on) Sequential Linear Programming Algorithms

3 Numerical Results
Optimization in Gas Transmission Networks

1. Optimization in Gas Transmission Networks

2. (A twist on) Sequential Linear Programming Algorithms

3. Numerical Results
GANESO™: Gas Networks Simulation and Optimization

A two-step SLP for MINLP problems
GANESO™: Gas Networks Simulation and Optimization

GANESO™ is a software developed by researchers at USC and ITMATI for Reganosa Company.
GANESO™: Gas Networks Simulation and Optimization

- GANESO™ is a software developed by researchers at USC and ITMATIC for Reganosa Company
- Ongoing project that started in 2011; around 15 researchers have participated.
GANESO™: Gas Networks Simulation and Optimization

- GANESO™ 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®: Gas Networks Simulation and Optimization

- **GANESO®** 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®**:  
  - 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® by end-user)
GANESO™: Gas Networks Simulation and Optimization

- **GANESO™** 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™**:
  - 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™ by end-user)
GANESO™: Gas Networks Simulation and Optimization

- GANESO™ 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™:
  - 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™ by end-user)

Nonlinear optimization
GANESO™: Gas Networks Simulation and Optimization

- GANESO™ 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™:
  - 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™ by end-user)

Nonlinear optimization
GANESOTM: Gas Networks Simulation and Optimization

- **GANESOTM** 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 **GANESOTM**:
  - 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 GANESOTM by end-user)

Nonlinear optimization

Stochastic programming

A two-step SLP for MINLP problems
GANESO™: Gas Networks Simulation and Optimization

- **GANESO™** 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™**:
  - 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™ by end-user)

Nonlinear optimization
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

Flow conservation constraints

\[ \sum_{k \in A} q_{ik}^{ini} - \sum_{k \in A} q_{ik}^{fin} = c_i \quad \forall i \in N \]

Demand nodes

\[ 0 \leq \sum_{k \in A} q_{ik}^{ini} - \sum_{k \in A} q_{ik}^{fin} \leq s_i \quad \forall i \in N \]

Supply nodes

Box Constraints

\[ \bar{q}_k \leq q_{ik} \leq \bar{q}_k \]

\[ \bar{p}_i^2 \leq p_{i}^2 \leq \bar{p}_i^2 \]

\[ \forall i \in N \]

Variables of the optimization problem

Flow through each pipe

Pressure at each node
Network flow problem

Flow conservation constraints

\[ \sum_{k \in A_{i}^{\text{ini}}} q_k - \sum_{k \in A_{i}^{\text{fin}}} q_k = c_i \]
\[ \forall i \in N^C \text{ demand nodes} \]

\[ 0 \leq \sum_{k \in A_{i}^{\text{ini}}} q_k - \sum_{k \in A_{i}^{\text{fin}}} q_k \leq s_i \]
\[ \forall i \in N^S \text{ supply nodes} \]
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 \text{ 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 \text{ supply nodes}

Box Constraints

\[ q_k \leq q_k \leq \bar{q}_k \]

\forall k \in A \text{ flow bounds}
Network flow problem

Flow conservation constraints
\[ \sum_{k \in A_i^{ini}} q_k - \sum_{k \in A_i^{fin}} q_k = c_i \]
\[ \forall i \in N^C \text{ demand nodes} \]

\[ 0 \leq \sum_{k \in A_i^{ini}} q_k - \sum_{k \in A_i^{fin}} q_k \leq s_i \]
\[ \forall i \in N^S \text{ supply nodes} \]

Box Constraints
\[ q_k \leq q_k \leq \bar{q}_k \]
\[ \forall k \in A \text{ flow bounds} \]

\[ p_i^2 \leq p_i^2 \leq \bar{p}_i^2 \]
\[ \forall i \in N \text{ pressure bounds} \]
**Network flow problem**

**Flow conservation constraints**
\[
\sum_{k \in A_i^{\text{ini}}} q_k - \sum_{k \in A_i^{\text{fin}}} q_k = c_i \\
\forall i \in N^C \text{ demand nodes}
\]

\[
0 \leq \sum_{k \in A_i^{\text{ini}}} q_k - \sum_{k \in A_i^{\text{fin}}} q_k \leq s_i \\
\forall i \in N^S \text{ supply nodes}
\]

**Box Constraints**
\[
q_k \leq q_k \leq \bar{q}_k \\
\forall k \in A \text{ flow bounds}
\]

\[
\bar{p}_i^2 \leq p_i^2 \leq \bar{p}_i^2 \\
\forall i \in N \text{ pressure bounds}
\]

**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

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

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

\[ p_i^2 - p_j^2 = \frac{16 L_k \lambda_k}{\pi^2 D_k^5} Z(p_m, T_m) RT_m |q_{ij}| q_{ij} + \frac{2g}{RT_m} \frac{p_i^2 + p_j^2}{2Z(p_m, T_m)} (h_j - h_i) \]
Gass loss equations

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

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

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

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

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

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

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**
\[
\begin{align*}
  p_i^2 &\leq p_i^2 \leq \bar{p}_i^2 & \forall i \in N & \text{pressure bounds} \\
  q_k &\leq q_k \leq \bar{q}_k & \forall k \in A & \text{flow bounds}
\end{align*}
\]

**Flow conservation constraints**
\[
\begin{align*}
  \sum_{k \in A_i^{ini}} q_k - \sum_{k \in A_i^{fin}} q_k &= c_i & \forall i \in N^C & \text{flow conservation at demand nodes} \\
  0 &\leq \sum_{k \in A_i^{ini}} q_k - \sum_{k \in A_i^{fin}} q_k &\leq s_i & \forall i \in N^S & \text{flow conservation at supply nodes}
\end{align*}
\]

**Gas loss constraints**
\[
\begin{align*}
  p_i^2 - p_j^2 &= \frac{16L_k \lambda_k}{\pi^2 D^5_k} Z(p_m, T_m) RT_m |q_k| q_k + \forall k \in A^n & \text{gas loss (}\lambda_k\text{ Weymouth)} \\
  + \frac{2g}{RT_m} \frac{p_i^2 + p_j^2}{2Z(p_m, T_m)} (h_j - h_i) & \text{height difference term}
\end{align*}
\]

**Gas consumption constraints**
\[
\begin{align*}
  g_k &= \frac{1}{e_h H_c} \frac{\gamma}{\gamma - 1} Z(p_m, T_{in}) RT_{in} \left( \frac{p_j}{p_i} \right)^{\frac{\gamma - 1}{\gamma}} - 1 \right) q_k
\end{align*}
\]

A two-step SLP for MINLP problems
Nonlinear nonconvex optimization problem (continuous)

**Obj. Function:** \[ \min \sum_{k \in A^c} g_k \]

**Box Constraints**
- \[ p_i^2 \leq p_i^2 \leq \bar{p}_i^2 \quad \forall i \in N \text{ pressure bounds} \]
- \[ q_k \leq q_k \leq \bar{q}_k \quad \forall k \in A \text{ flow bounds} \]

**Flow conservation constraints**
- \[ \sum_{k \in A^{ini}_i} q_k - \sum_{k \in A^{fin}_i} q_k = c_i \quad \forall i \in N^C \text{ flow conservation at demand nodes} \]
- \[ 0 \leq \sum_{k \in A^{ini}_i} q_k - \sum_{k \in A^{fin}_i} q_k \leq s_i \quad \forall i \in N^S \text{ flow conservation at supply nodes} \]

**Gas loss constraints**
- \[ p_i^2 - p_j^2 = \frac{16 L_k \lambda_k}{\pi^2 D^5_k} Z(p_m, T_m) RT_m |q_k| q_k + \frac{2g}{RT_m} \frac{p_i^2 + p_j^2}{2Z(p_m, T_m)} (h_j - h_i) \quad \forall k \in A^n \text{ gas loss (\lambda_k Weymouth)} \]

**Height difference term**

**Gas consumption constraints**
- \[ g_k = \frac{1}{e_h H_c} \frac{\gamma}{\gamma - 1} Z(p_m, T_{in}) RT_{in} \left( \left( \frac{p_j}{p_i} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right) q_k \]

A two-step SLP for MINLP problems

RSME 2017
Size and complexity of real instances
Size and complexity of real instances

Spanish primary gas network
- \( \approx 1000 \) variables (\( \approx 500 \) pipes and \( \approx 500 \) nodes)
- \( \approx 1000 \) constraints (and \( \approx 2000 \) box constraints)
- \( \approx 500 \) constraints are \textit{nonlinear}
Size and complexity of real instances

Spanish primary gas network

- $\approx 1000$ variables ($\approx 500$ pipes and $\approx 500$ nodes)
- $\approx 1000$ constraints (and $\approx 2000$ box constraints)
- $\approx 500$ constraints are nonlinear

To be solved routinely by the company
(A twist on) Sequential Linear Programming Algorithms

1. Optimization in Gas Transmission Networks

2. (A twist on) Sequential Linear Programming Algorithms

3. Numerical Results
Approaches to solve the problem

Spanish primary gas network

- \( \approx 1000 \) variables (\( \approx 500 \) pipes and \( \approx 500 \) nodes)
- \( \approx 1000 \) constraints (and \( \approx 2000 \) box constraints)
- \( \approx 500 \) constraints are nonlinear
Approaches to solve the problem

Spanish primary gas network
- $\approx 1000$ variables ($\approx 500$ pipes and $\approx 500$ nodes)
- $\approx 1000$ constraints (and $\approx 2000$ box constraints)
- $\approx 500$ 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 $\Rightarrow$ piecewise linear functions + integer variables
Approaches to solve the problem

Spanish primary gas network

- $\approx 1000$ variables ($\approx 500$ pipes and $\approx 500$ nodes)
- $\approx 1000$ constraints (and $\approx 2000$ box constraints)
- $\approx 500$ constraints are nonlinear

How to solve this problem?

- **Global optimization algorithms** on approximations of the problem (cannot handle real-size problems)
  \[
  \text{nonlinearities} \Rightarrow \text{piecewise linear functions} + \text{integer variables}
  \]
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 $\Rightarrow$ 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 (No Trust Region)**
- **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 (No Trust Region)**
  - 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 (No Trust Region)**
  - 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

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \\
\text{inequality constraints} & \quad g_i(x) \leq 0, \quad i = 1, \ldots, m \\
\text{equality constrains} & \quad h_j(x) = 0, \quad j = 1, \ldots, l \\
\text{linear constraints} & \quad x \in X = \{x \in \mathbb{R}^n : Ax \leq b\}
\end{align*}
\]

where \( f, g_i \) and \( h_j \) are nonlinear functions.
SLP-NTR (No Trust Region)

Classic SLP
SLP-NTR (No Trust Region)

Classic SLP

- At iteration $k$ we have a candidate solution $x^k$
SLP-NTR (No Trust Region)

Classic SLP

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

$$\minimize \nabla f(x^k) \text{ subject to } \begin{align*}
\text{inequality constraints: } g_i(x^k) + \nabla g_i(x^k)(x - x^k) &\leq 0 \\
\text{equality constraints: } h_j(x^k) + \nabla h_j(x^k)(x - x^k) &= 0 \\
\text{linear constraints: } x \in X = \{x \in \mathbb{R}^n : Ax \leq b\} 
\end{align*}$$

Hard to accommodate binary variables with the trust region
**SLP-NTR (No Trust Region)**

**Classic SLP**

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

  \[
  \text{minimize} \quad \nabla f(x^k)^t x \\
  \text{subject to} \\
  \text{inequality constraints} \quad g_i(x^k) + \nabla g_i(x^k)^t (x - x^k) \leq 0 \quad i = 1, \ldots, m \\
  \text{equality constraints} \quad h_j(x^k) + \nabla h_j(x^k)^t (x - x^k) = 0 \quad j = 1, \ldots, l \\
  \text{linear constraints} \quad x \in X = \{x \in \mathbb{R}^n : Ax \leq b\} \\
  \text{trust region} \quad -d_k \leq x - x^k \leq d_k \]
**SLP-NTR (No Trust Region)**

**Classic SLP**

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

  \[
  \begin{align*}
  \text{minimize} & \quad \nabla f(x^k)^t x \\
  \text{subject to} & \\
  \text{inequality constraints} & \quad g_i(x^k) + \nabla g_i(x^k)^t (x - x^k) \leq 0 \quad i = 1, \ldots, m \\
  \text{equality constraints} & \quad h_j(x^k) + \nabla h_j(x^k)^t (x - x^k) = 0 \quad j = 1, \ldots, l \\
  \text{linear constraints} & \quad x \in X = \{x \in \mathbb{R}^n : Ax \leq b\} \\
  \text{trust region} & \quad -d_k \leq x - x^k \leq d_k
  \end{align*}
  \]
SLP-NTR  (No Trust Region)

Classic SLP

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

\[
\begin{align*}
\text{minimize} & \quad \nabla f(x^k)^t x \\
\text{subject to} & \\
\text{inequality constraints} & \quad g_i(x^k) + \nabla g_i(x^k)^t (x - x^k) \leq 0 \quad i = 1, \ldots, m \\
\text{equality constraints} & \quad h_j(x^k) + \nabla h_j(x^k)^t (x - x^k) = 0 \quad j = 1, \ldots, l \\
\text{linear constraints} & \quad x \in X = \{ x \in \mathbb{R}^n : Ax \leq b \} \\
\text{trust region} & \quad -d_k \leq x - x^k \leq d_k
\end{align*}
\]

Hard to accommodate binary variables with the trust region
SLP-NTR (No Trust Region)

SLP-NTR

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

\[
\begin{align*}
\text{minimize} & \quad \nabla f(x^k)^t x \\
\text{subject to} & \quad \text{inequality constraints} \quad g_i(x^k) + \nabla g_i(x^k)^t (x - x^k) \leq 0 \quad i = 1, \ldots, m \\
& \quad \text{equality constraints} \quad h_j(x^k) + \nabla h_j(x^k)^t (x - x^k) = 0 \quad j = 1, \ldots, l \\
& \quad \text{linear constraints} \quad x \in X = \{x \in \mathbb{R}^n : Ax \leq b\} \\
& \quad \text{trust region} \quad -d_k \leq x - x^k \leq d_k
\end{align*}
\]
SLP-NTR (No Trust Region)

SLP-NTR

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

  \[
  \begin{align*}
  \text{minimize} & \quad \nabla f(x^k)^t x \\
  \text{subject to} & \quad g_i(x^k) + \nabla g_i(x^k)^t (x - x^k) \leq 0 \quad i = 1, \ldots, m \\
  & \quad h_j(x^k) + \nabla h_j(x^k)^t (x - x^k) = 0 \quad j = 1, \ldots, l \\
  & \quad x \in X = \{x \in \mathbb{R}^n : Ax \leq b\} \\
  \end{align*}
  \]

- We remove the constraints that define the trust region
**SLP-NTR (No Trust Region)**

**SLP-NTR**

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

  \[
  \text{minimize} \quad \nabla f(x^k)^t x \\
  \text{subject to} \\
  \begin{align*}
  \text{inequality constraints} & \quad g_i(x^k) + \nabla g_i(x^k)^t (x - x^k) \leq 0 \quad i = 1, \ldots, m \\
  \text{equality constraints} & \quad h_j(x^k) + \nabla h_j(x^k)^t (x - x^k) = 0 \quad j = 1, \ldots, l \\
  \text{linear constraints} & \quad x \in X = \{x \in \mathbb{R}^n : Ax \leq b\} \\
  \text{trust region} & \quad ||x^k - d_k|| \leq x - x^k \leq ||x^k + d_k||
  \end{align*}
  \]

- 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 $x^k$
- We solve the linearization of NLP about $x^k$, LP($x^k$):

\[
\begin{align*}
\text{minimize} & \quad \nabla f(x^k)^t x \\
\text{subject to} & \\
\text{inequality constraints} & \quad g_i(x^k) + \nabla g_i(x^k)^t (x - x^k) \leq 0 \quad i = 1, \ldots, m \\
\text{equality constraints} & \quad h_j(x^k) + \nabla h_j(x^k)^t (x - x^k) = 0 \quad j = 1, \ldots, l \\
\text{linear constraints} & \quad x \in X = \{ x \in \mathbb{R}^n : Ax \leq b \}
\end{align*}
\]

- 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)

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.
  \[ x \in [-1, 1] \times [-1, 1] \] (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 \( \rightarrow \) 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
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 (No Trust Region)
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
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
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
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 ($\min_{x \in [-1, 1]} x^2$)
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)
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)
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 \(\{x^k\}\)
  are sufficiently close \(\rightarrow\) 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 (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 \(\{x^k\}\) are sufficiently close \(\rightarrow\) 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 (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 $\{x^k\}$ are sufficiently close $\rightarrow$ 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 (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 \(\{x^k\}\)
   are sufficiently close \(\rightarrow\) 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 \(\{x^k\}\) are sufficiently close \(\rightarrow\) 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 (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 $\{x^k\}$ are sufficiently close $\rightarrow$ 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 (No Trust Region)
- Step 2. Classic SLP
Summary (algorithm for MINLP problems)

SLP: SLP-NTR + Classic SLP

- Step 1. SLP-NTR (No Trust Region)
- Step 2. Classic SLP

Features of our two-step approach
Summary (algorithm for MINLP problems)

2SLP: SLP-NTR + Classic SLP

- Step 1. SLP-NTR (No Trust Region)
- 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 (**No Trust Region**)
- **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 (No Trust Region)
- 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** *(No Trust Region)*
- **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 (No Trust Region)**
- **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

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
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™ user interface

Optimization in Gas Transmission Networks (A twist on) SLP Algorithms Numerical Results

A two-step SLP for MINLP problems RSME 2017
GANESO™ user interface

Interactive!!

A two-step SLP for MINLP problems

RSME 2017
GANESO™ user interface

Interactive!!

Routinely used by the company

Optimization in Gas Transmission Networks

(A twist on) SLP Algorithms

Numerical Results

A two-step SLP for MINLP problems

RSME 2017
Numerical Results

1 Optimization in Gas Transmission Networks

2 (A twist on) Sequential Linear Programming Algorithms

3 Numerical Results
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)

- 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)

- 75 NLP real size instances ≈ 1000 variables and constraints
- 200 iterations maximum
- Algorithm convergence:
  - CSLP: 100%
  - SLP-NTR: 66%
  - 2SLP: 100%
  - 25% 2SLP outperforms CSLP
  - 5% CSLP outperforms 2SLP
Tests on the Spanish Gas Transmission Network (NLP)

- 75 NLP real size instances
  ≈ 1000 variables and constraints

- 200 iterations maximum
- Algorithm convergence:
  - CSLP: 100%
  - SLP-NTR: 66%
  - 2SLP: 100%

- 25% 2SLP outperforms CSLP
- 5% CSLP outperforms 2SLP
Tests on the Spanish Gas Transmission Network (NLP)

- 75 NLP real size instances
- ≈ 1000 variables and constraints
- 200 iterations maximum
- Algorithm convergence:
  - CSLP: 100%
  - SLP-NTR: 66%
  - 2SLP: 100%
- 25% 2SLP outperforms CSLP
- 5% CSLP outperforms 2SLP
Tests on the Spanish Gas Transmission Network (NLP)

- 75 NLP real size instances
  ≈ 1000 variables and constraints

- 200 iterations maximum

- Algorithm convergence:
  - CSLP: 100%
  - SLP-NTR: 66%
  - 2SLP: 100%

- 25% 2SLP outperforms CSLP
- 5% CSLP outperforms 2SLP
Tests on the Spanish Gas Transmission Network (NLP)

- 75 NLP real size instances
  \( \approx 1000 \) variables and constraints

Algorithm convergence:
- CSLP: 100%
- SLP-NTR: 66%
- 2SLP: 100%

25% 2SLP outperforms CSLP
5% CSLP outperforms 2SLP
Tests on the Spanish Gas Transmission Network (NLP)

- 75 NLP real size instances
- \( \approx 1000 \) variables and constraints
- 200 iterations maximum

Algorithm convergence:
- CSLP: 100%
- SLP-NTR: 66%
- 2SLP: 100%

25% 2SLP outperforms CSLP
5% CSLP outperforms 2SLP
Tests on the Spanish Gas Transmission Network (NLP)

- 75 NLP real size instances
- \(\approx 1000\) variables and constraints
- 200 iterations maximum
- Algorithm convergence:
  - CSLP: 100%
  - SLP-NTR: 66%
  - 2SLP: 100%

\(25\%\) 2SLP outperforms CSLP
\(5\%\) CSLP outperforms 2SLP
Tests on the Spanish Gas Transmission Network (NLP)

- 75 NLP real size instances
- \( \approx 1000 \) variables and constraints
- 200 iterations maximum
- Algorithm convergence:
  - CSLP: 100%
  - SLP-NTR: 66%
  - 2SLP: 100%
  - 25% 2SLP outperforms CSLP
  - 5% CSLP outperforms 2SLP
Tests on the Spanish Gas Transmission Network (NLP)

- CSLP
- SLP-NTR
- 2SLP

- 75 real size instances
- \( \approx 1000 \) variables and constraints

- 200 iterations maximum

- Running times:
  - SLP-NTR \( \ll \) 2SLP \( \ll \) CSLP

- 2SLP shows superior performance
Tests on the Spanish Gas Transmission Network (NLP)

- CSLP
- SLP-NTR
- 2SLP

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

Running times:
- SLP-NTR ≪ 2SLP ≪ CSLP

2SLP superior performance
Tests on the Spanish Gas Transmission Network (NLP)

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

Running times:
- CSLP ≪ SLP-NTR ≪ 2SLP

2SLP superior performance
Tests on the Spanish Gas Transmission Network (NLP)

- 75 NLP real size instances
- \( \approx 1000 \) variables and constraints
- 200 iterations maximum
- Running times:
  - SLP-NTR \( \ll \) 2SLP \( \ll \) CSLP

![Graph showing computational time distribution for different algorithms: CSLP, SLP-NTR, 2SLP.](image)
Tests on the Spanish Gas Transmission Network (NLP)

- 75 NLP real size instances
- ≈ 1000 variables and constraints
- 200 iterations maximum
- Running times:
  - SLP-NTR ≪ 2SLP ≪ CSLP
- 2SLP superior performance
Tests on the Spanish Gas Transmission Network (MINLP)

Next task.
Designing a full set of test instances
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 Spanish Gas Transmission Network (MINLP)

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

![Box plot comparing 2SLP and CSLP-Enumeration](graph.png)
Tests on the Spanish Gas Transmission Network (MINLP)

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

![Box plots comparing Objective function and Computational time for 2SLP and 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: \( \approx 50 \) 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: \( \approx 50 \) 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

<table>
<thead>
<tr>
<th>NLP problem</th>
<th>CSLP</th>
<th>SLP-NTR</th>
<th>2SLP</th>
<th>BARON</th>
<th>Knitro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective function</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
</tr>
<tr>
<td>Computational time</td>
<td>0.3654</td>
<td>0.3570</td>
<td>0.3570</td>
<td>0.0733</td>
<td>0.0093</td>
</tr>
</tbody>
</table>

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

NLP formulation of the problem

<table>
<thead>
<tr>
<th>NLP problem</th>
<th>CSLP</th>
<th>SLP-NTR</th>
<th>2SLP</th>
<th>BARON</th>
<th>Knitro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective function</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
</tr>
<tr>
<td>Computational time</td>
<td>0.3654</td>
<td>0.3570</td>
<td>0.3570</td>
<td>0.0733</td>
<td>0.0093</td>
</tr>
</tbody>
</table>

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: ≈ 50 variables and constraints

NLP formulation of the problem

<table>
<thead>
<tr>
<th>NLP problem</th>
<th>CSLP</th>
<th>SLP-NTR</th>
<th>2SLP</th>
<th>BARON</th>
<th>Knitro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective function</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
</tr>
<tr>
<td>Computational time</td>
<td>0.3654</td>
<td>0.3570</td>
<td>0.3570</td>
<td>0.0733</td>
<td>0.0093</td>
</tr>
</tbody>
</table>

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: \( \approx 50 \) variables and constraints

### NLP formulation of the problem

<table>
<thead>
<tr>
<th>NLP problem</th>
<th>CSLP</th>
<th>SLP-NTR</th>
<th>2SLP</th>
<th>BARON</th>
<th>Knitro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective function</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
</tr>
<tr>
<td>Computational time</td>
<td>0.3654</td>
<td>0.3570</td>
<td>0.3570</td>
<td>0.0733</td>
<td>0.0093</td>
</tr>
</tbody>
</table>

### 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 \( q_{ij} \)
- \( \approx 25 \) binary variables and 50 additional constraints

<table>
<thead>
<tr>
<th>NLP problem</th>
<th>2SLP</th>
<th>BARON</th>
<th>Knitro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective function</td>
<td>91.0562</td>
<td>91.0562</td>
<td>94.8715 (infeasible)</td>
</tr>
<tr>
<td>Computational time</td>
<td>0.6497</td>
<td>231.2602</td>
<td>0.0983</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>NLP problem</th>
<th>CSLP</th>
<th>SLP-NTR</th>
<th>2SLP</th>
<th>BARON</th>
<th>Knitro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective function</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
</tr>
<tr>
<td>Computational time</td>
<td>0.3654</td>
<td>0.3570</td>
<td>0.3570</td>
<td>0.0733</td>
<td>0.0093</td>
</tr>
</tbody>
</table>

### 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 $q_{ij}$
- $\approx 25$ binary variables and 50 additional constraints

<table>
<thead>
<tr>
<th>NLP problem</th>
<th>2SLP</th>
<th>BARON</th>
<th>Knitro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective function</td>
<td>91.0562</td>
<td>91.0562</td>
<td>94.8715    (infeasible)</td>
</tr>
<tr>
<td>Computational time</td>
<td>0.6497</td>
<td>231.2602</td>
<td>0.0983</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>NLP problem</th>
<th>CSLP</th>
<th>SLP-NTR</th>
<th>2SLP</th>
<th>BARON</th>
<th>Knitro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective function</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
</tr>
<tr>
<td>Computational time</td>
<td>0.3654</td>
<td>0.3570</td>
<td>0.3570</td>
<td>0.0733</td>
<td>0.0093</td>
</tr>
</tbody>
</table>

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 \( q_{ij} \)
- \( \approx 25 \) binary variables and 50 additional constraints

<table>
<thead>
<tr>
<th>NLP problem</th>
<th>2SLP</th>
<th>BARON</th>
<th>Knitro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective function</td>
<td>91.0562</td>
<td>91.0562</td>
<td>94.8715 (infeasible)</td>
</tr>
<tr>
<td>Computational time</td>
<td>0.6497</td>
<td>231.2602</td>
<td>0.0983</td>
</tr>
</tbody>
</table>
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**

<table>
<thead>
<tr>
<th>NLP problem</th>
<th>CSLP</th>
<th>SLP-NTR</th>
<th>2SLP</th>
<th>BARON</th>
<th>Knitro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective function</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
<td>91.0562</td>
</tr>
<tr>
<td>Computational time</td>
<td>0.3654</td>
<td>0.3570</td>
<td>0.3570</td>
<td>0.0733</td>
<td>0.0093</td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>NLP problem</th>
<th>2SLP</th>
<th>BARON</th>
<th>Knitro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective function</td>
<td>91.0562</td>
<td>91.0562</td>
<td>94.8715 (infeasible)</td>
</tr>
<tr>
<td>Computational time</td>
<td>0.6497</td>
<td>231.2602</td>
<td>0.0983</td>
</tr>
</tbody>
</table>

**Next task.** Designing a full set of test instances
Tests on multicommodity flow problems (NLP)

- Linear constraints and \textit{nonlinear} objective function
Tests on multicommodity flow problems (NLP)

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

<table>
<thead>
<tr>
<th>Problem</th>
<th>N</th>
<th>E</th>
<th>T</th>
<th>Constr.</th>
<th>Variab.</th>
<th>z_{opt}</th>
<th>Relative error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planar problems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSLP</td>
<td>30</td>
<td>150</td>
<td>92</td>
<td>2760</td>
<td>13800</td>
<td>4.445 × 10^{-7}</td>
<td>0.0074</td>
</tr>
<tr>
<td>SLP-NTR</td>
<td>30</td>
<td>150</td>
<td>92</td>
<td>2760</td>
<td>13800</td>
<td>4.445 × 10^{-7}</td>
<td>0.0074</td>
</tr>
<tr>
<td>2SLP</td>
<td>30</td>
<td>150</td>
<td>92</td>
<td>2760</td>
<td>13800</td>
<td>4.445 × 10^{-7}</td>
<td>0.0074</td>
</tr>
<tr>
<td>Grid problems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G1</td>
<td>25</td>
<td>80</td>
<td>50</td>
<td>1250</td>
<td>4000</td>
<td>8.336 × 10^{-5}</td>
<td>0.0003</td>
</tr>
<tr>
<td>G2</td>
<td>25</td>
<td>80</td>
<td>100</td>
<td>2500</td>
<td>8000</td>
<td>1.727 × 10^{-6}</td>
<td>0.0006</td>
</tr>
<tr>
<td>G3</td>
<td>100</td>
<td>360</td>
<td>50</td>
<td>5000</td>
<td>18000</td>
<td>5.320 × 10^{-6}</td>
<td>0.0000</td>
</tr>
<tr>
<td>G4</td>
<td>100</td>
<td>360</td>
<td>100</td>
<td>10000</td>
<td>36000</td>
<td>5.055 × 10^{-6}</td>
<td>0.0000</td>
</tr>
<tr>
<td>G5</td>
<td>225</td>
<td>840</td>
<td>100</td>
<td>22500</td>
<td>84000</td>
<td>5.079 × 10^{-6}</td>
<td>0.0000</td>
</tr>
<tr>
<td>G6</td>
<td>225</td>
<td>840</td>
<td>200</td>
<td>45000</td>
<td>168000</td>
<td>6.051 × 10^{-7}</td>
<td>0.0001</td>
</tr>
<tr>
<td>G7</td>
<td>400</td>
<td>1520</td>
<td>400</td>
<td>160000</td>
<td>608000</td>
<td>6.079 × 10^{-7}</td>
<td>0.0000</td>
</tr>
<tr>
<td>Telecommunication-like problems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N22</td>
<td>14</td>
<td>22</td>
<td>23</td>
<td>322</td>
<td>506</td>
<td>8.710 × 10^{-3}</td>
<td>0.0131</td>
</tr>
<tr>
<td>N148</td>
<td>58</td>
<td>148</td>
<td>122</td>
<td>7076</td>
<td>18056</td>
<td>4.020 × 10^{-5}</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

All approaches very competitive in terms of objective function.
Tests on multicommodity flow problems (NLP)

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

<table>
<thead>
<tr>
<th>Problem</th>
<th>N</th>
<th>E</th>
<th>T</th>
<th>Constr.</th>
<th>Variab.</th>
<th>z_{opt}</th>
<th>Relative error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planar problems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSLP</td>
<td>SLP-NTR</td>
<td>2SLP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P30</td>
<td>30</td>
<td>150</td>
<td>92</td>
<td>2760</td>
<td>13800</td>
<td>4.445 × 10^{-7}</td>
<td>0.0074</td>
</tr>
<tr>
<td>P50</td>
<td>50</td>
<td>250</td>
<td>267</td>
<td>13350</td>
<td>66750</td>
<td>1.212 × 10^{-8}</td>
<td>0.0202</td>
</tr>
<tr>
<td>P80</td>
<td>80</td>
<td>440</td>
<td>543</td>
<td>43440</td>
<td>238920</td>
<td>1.819 × 10^{-8}</td>
<td>0.0174</td>
</tr>
<tr>
<td>P100</td>
<td>100</td>
<td>532</td>
<td>1085</td>
<td>108500</td>
<td>577220</td>
<td>2.291 × 10^{-8}</td>
<td>0.0212</td>
</tr>
<tr>
<td>Grid problems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G1</td>
<td>25</td>
<td>80</td>
<td>50</td>
<td>1250</td>
<td>4000</td>
<td>8.336 × 10^{-5}</td>
<td>0.0003</td>
</tr>
<tr>
<td>G2</td>
<td>25</td>
<td>80</td>
<td>100</td>
<td>2500</td>
<td>8000</td>
<td>1.727 × 10^{-6}</td>
<td>0.0006</td>
</tr>
<tr>
<td>G3</td>
<td>100</td>
<td>360</td>
<td>50</td>
<td>5000</td>
<td>18000</td>
<td>1.532 × 10^{-6}</td>
<td>0.0000</td>
</tr>
<tr>
<td>G4</td>
<td>100</td>
<td>360</td>
<td>100</td>
<td>10000</td>
<td>36000</td>
<td>3.055 × 10^{-6}</td>
<td>0.0000</td>
</tr>
<tr>
<td>G5</td>
<td>225</td>
<td>840</td>
<td>100</td>
<td>22500</td>
<td>84000</td>
<td>5.079 × 10^{-6}</td>
<td>0.0000</td>
</tr>
<tr>
<td>G6</td>
<td>225</td>
<td>840</td>
<td>200</td>
<td>45000</td>
<td>168000</td>
<td>1.051 × 10^{-6}</td>
<td>0.0001</td>
</tr>
<tr>
<td>G7</td>
<td>400</td>
<td>1520</td>
<td>400</td>
<td>160000</td>
<td>608000</td>
<td>2.607 × 10^{-7}</td>
<td>0.0000</td>
</tr>
<tr>
<td>Telecommunication-like problems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N22</td>
<td>14</td>
<td>22</td>
<td>23</td>
<td>322</td>
<td>506</td>
<td>1.871 × 10^{-3}</td>
<td>0.0131</td>
</tr>
<tr>
<td>N148</td>
<td>58</td>
<td>148</td>
<td>122</td>
<td>7076</td>
<td>18056</td>
<td>4.022 × 10^{-5}</td>
<td>0.0000</td>
</tr>
<tr>
<td>Transportation problems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-F</td>
<td>24</td>
<td>76</td>
<td>528</td>
<td>12672</td>
<td>40128</td>
<td>3.202 × 10^{-5}</td>
<td>0.0050</td>
</tr>
</tbody>
</table>
Tests on multicommodity flow problems (NLP)

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

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

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

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

- All approaches very competitive in terms of objective function
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.
Tests on multicommodity flow problems (NLP)

- Now CSLP is the fastest one

Diagram: Density of computational time for different algorithms (CSLP, SLP-NTR, 2SLP)
Tests on multicommodity flow problems (NLP)

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

![Graph showing computational time vs. density for different algorithms: CSLP, SLP-NTR, 2SLP. CSLP is the green line and appears to have the lowest computational time. The graph suggests that CSLP is more efficient for solving these problems.](image-url)
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)

February 3rd, 2017