

Algorithms for mathematical programs with complementarity constraints

Summer School on Nonlinear Optimization with Mixed Integer and Complementarity
Constraints – NOMICC26

Moritz Diehl, Andrea Ghezzi, Sven Leyffer, and Armin Nurkanović

University of Freiburg, Argonne National Laboratory

September 30, 2026

Outline of the lecture

1. Overview of MPCC methods

2. Regularization and penalty methods

Regularization methods

Penalty methods

3. Combinatorial and active-set methods

A sequential linearization method for bound-constrained MPCCs

A sequential LPCC-EQP method

MPECopt: an algorithm for computing a B-stationary points

Sequential quadratic programming with complementarity constraints

Outline of the lecture

1. Overview of MPCC methods

2. Regularization and penalty methods

Regularization methods

Penalty methods

3. Combinatorial and active-set methods

A sequential linearization method for bound-constrained MPCCs

A sequential LPCC–EQP method

MPECopt: an algorithm for computing a B-stationary points

Sequential quadratic programming with complementarity constraints

Numerical methods for MPCCs

MPCC

$$\min_{x \in \mathbb{R}^n} f(x) \quad (1a)$$

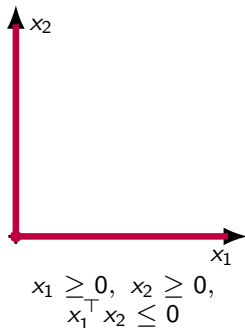
$$\text{s.t.} \quad g(x) = 0, \quad (1b)$$

$$h(x) \geq 0, \quad (1c)$$

$$0 \leq x_1 \perp x_2 \geq 0. \quad (1d)$$

$$x = (x_0, x_1, x_2) \in \mathbb{R}^n, \quad x_0 \in \mathbb{R}^p, \quad x_1, x_2 \in \mathbb{R}^m,$$

- ▶ Standard NLP methods solve the KKT conditions.
- ▶ MPCCs violate constraint qualifications, and the KKT conditions may not be necessary.
- ▶ There are many stationary concepts for MPCCs, and not all are useful.



Numerical methods for MPCCs

MPCC

$$\min_{x \in \mathbb{R}^n} f(x) \quad (1a)$$

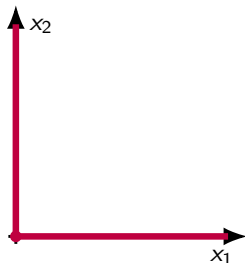
$$\text{s.t.} \quad g(x) = 0, \quad (1b)$$

$$h(x) \geq 0, \quad (1c)$$

$$0 \leq x_1 \perp x_2 \geq 0. \quad (1d)$$

$$x = (x_0, x_1, x_2) \in \mathbb{R}^n, \quad x_0 \in \mathbb{R}^p, \quad x_1, x_2 \in \mathbb{R}^m,$$

- ▶ Standard NLP methods solve the KKT conditions.
- ▶ MPCCs violate constraint qualifications, and the KKT conditions may not be necessary.
- ▶ There are many stationary concepts for MPCCs, and not all are useful.
- ▶ **Workaround/main idea:** solve a (finite) sequence of more regular problems.



$$x_1 \geq 0, \quad x_2 \geq 0, \\ x_1^T x_2 \leq 0$$

Families of MPCC methods

MPCC solution methods

Continuous reformulations

Explicit disjunction

MPCC solution methods

Continuous reformulations

MPCC as NLP

Mechanism: one NLP reformulation

Pro: reuse NLP solvers

Con: ill-conditioning; poor scaling

Regularization methods

Mechanism: NLP sequence, $\sigma^k \downarrow 0$

Pro: easy to implement

Con: ill-conditioning; weaker theory

Explicit disjunction

MPCC solution methods

Continuous reformulations

MPCC as NLP

Mechanism: one NLP reformulation

Pro: reuse NLP solvers

Con: ill-conditioning; poor scaling

Regularization methods

Mechanism: NLP sequence, $\sigma^k \downarrow 0$

Pro: easy to implement

Con: ill-conditioning; weaker theory

Explicit disjunction

Active-set, pivoting

Mechanism: TNLP/BNLP sequence

Pro: strong guarantees

Con: combinatorial complexity

Mixed-integer

Mechanism: equivalent MINLP

Pro: simple formulation

Con: potentially very slow

MPCC solution methods

Continuous reformulations

MPCC as NLP

Mechanism: one NLP reformulation

Pro: reuse NLP solvers

Con: ill-conditioning; poor scaling

Regularization methods

Mechanism: NLP sequence, $\sigma^k \downarrow 0$

Pro: easy to implement

Con: ill-conditioning; weaker theory

Explicit disjunction

Active-set, pivoting

Mechanism: TNLP/BNLP sequence

Pro: strong guarantees

Con: combinatorial complexity

Mixed-integer

Mechanism: equivalent MINLP

Pro: simple formulation

Con: potentially very slow

Main distinction

Continuous methods hide the disjunction in an NLP and structure-exploiting methods search directly over its pieces.

Outline of the lecture

1. Overview of MPCC methods

2. Regularization and penalty methods

Regularization methods

Penalty methods

3. Combinatorial and active-set methods

A sequential linearization method for bound-constrained MPCCs

A sequential LPCC-EQP method

MPECopt: an algorithm for computing a B-stationary points

Sequential quadratic programming with complementarity constraints

Scholtes' global relaxation method

The easiest to implement and the most efficient regularization method [Scholtes, 2001].

$\text{Reg}(\sigma^k)$

$$\min_{x \in \mathbb{R}^n} f(x)$$

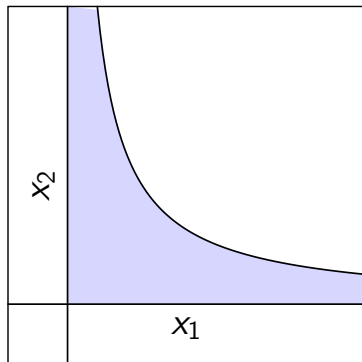
$$\text{s.t.} \quad g(x) = 0,$$

$$h(x) \geq 0,$$

$$x_1 \geq 0,$$

$$x_2 \geq 0,$$

$$x_{1,i}x_{2,i} \leq \sigma^k, \quad i = 1, \dots, m.$$



Scholtes' global relaxation method

The easiest to implement and the most efficient regularization method [Scholtes, 2001].

$\text{Reg}(\sigma^k)$

$$\min_{x \in \mathbb{R}^n} f(x)$$

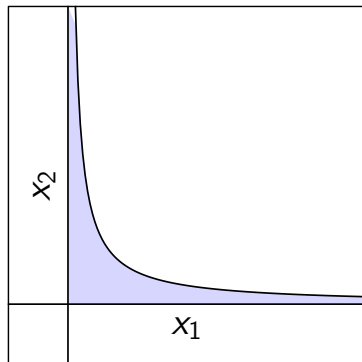
$$\text{s.t.} \quad g(x) = 0,$$

$$h(x) \geq 0,$$

$$x_1 \geq 0,$$

$$x_2 \geq 0,$$

$$x_{1,i}x_{2,i} \leq \sigma^k, \quad i = 1, \dots, m.$$



Scholtes' global relaxation method

The easiest to implement and the most efficient regularization method [Scholtes, 2001].

$\text{Reg}(\sigma^k)$

$$\min_{x \in \mathbb{R}^n} f(x)$$

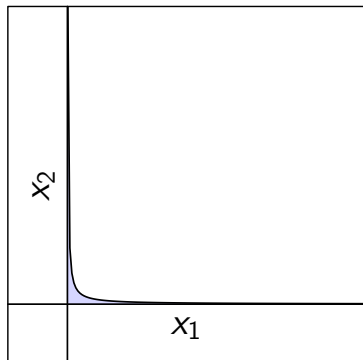
$$\text{s.t.} \quad g(x) = 0,$$

$$h(x) \geq 0,$$

$$x_1 \geq 0,$$

$$x_2 \geq 0,$$

$$x_{1,i}x_{2,i} \leq \sigma^k, \quad i = 1, \dots, m.$$



Scholtes' global relaxation method

The easiest to implement and the most efficient regularization method [Scholtes, 2001].

$\text{Reg}(\sigma^k)$

$$\min_{x \in \mathbb{R}^n} f(x)$$

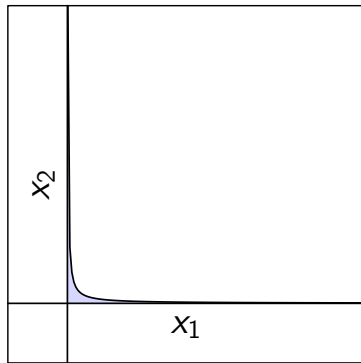
$$\text{s.t.} \quad g(x) = 0,$$

$$h(x) \geq 0,$$

$$x_1 \geq 0,$$

$$x_2 \geq 0,$$

$$x_{1,i}x_{2,i} \leq \sigma^k, \quad i = 1, \dots, m.$$



Theorem ([Scholtes, 2001, Hoheisel et al., 2013])

Let $\{\sigma^k\} \downarrow 0$ and let x^k be a stationary point of $\text{Reg}(\sigma^k)$ with $x^k \rightarrow x^*$ such that MPCC-MFCQ holds at x^* . Then x^* is a C-stationary point of the MPCC (1).

Smoothing the MPCC

Instead of relaxing $x_1, x_2 \geq 0$ $\text{diag}(x_1)x_2 \leq \sigma e$, use smoothing $x_1, x_2 \geq 0$, $\text{diag}(x_1)x_2 = \sigma e$.

RegEq(σ^k)

$$\min_{x \in \mathbb{R}^n} f(x)$$

s.t.

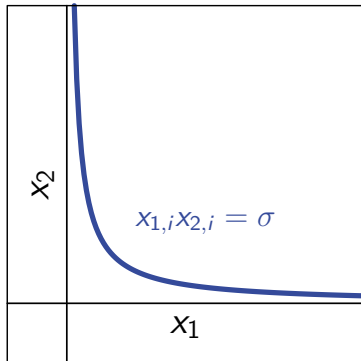
$$g(x) = 0,$$

$$h(x) \geq 0,$$

$$x_1 \geq 0,$$

$$x_2 \geq 0,$$

$$x_{1,i}x_{2,i} = \sigma^k, \quad i = 1, \dots, m.$$



Smoothing method have drawbacks

Usually benefit in using them. Have elaborate convergence theory [Ralph and Wright, 2004].

Consider the two-dimensional MPCC:

$$\begin{array}{ll} \min_{x \in \mathbb{R}^2} & (x_1 + 1)^2 + (x_2 + 1)^2 \\ \text{s.t.} & 0 \leq x_1 \perp x_2 \geq 0. \end{array}$$

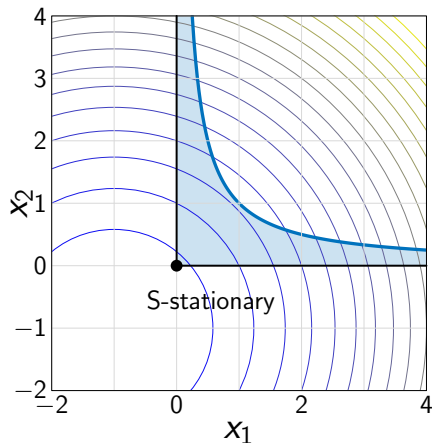
Smoothing method have drawbacks

Usually benefit in using them. Have elaborate convergence theory [Ralph and Wright, 2004].

Consider the two-dimensional MPCC:

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & (x_1 + 1)^2 + (x_2 + 1)^2 \\ \text{s.t.} \quad & 0 \leq x_1 \perp x_2 \geq 0. \end{aligned}$$

- Relaxation method converges in one iteration to $x^* = (0, 0)$.



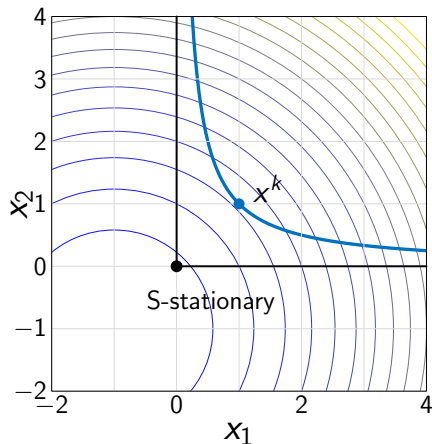
Smoothing method have drawbacks

Usually benefit in using them. Have elaborate convergence theory [Ralph and Wright, 2004].

Consider the two-dimensional MPCC:

$$\begin{array}{ll} \min_{x \in \mathbb{R}^2} & (x_1 + 1)^2 + (x_2 + 1)^2 \\ \text{s.t.} & 0 \leq x_1 \perp x_2 \geq 0. \end{array}$$

- ▶ Relaxation method converges in one iteration to $x^* = (0, 0)$.
- ▶ Smoothing needs an infinite sequence for convergence.



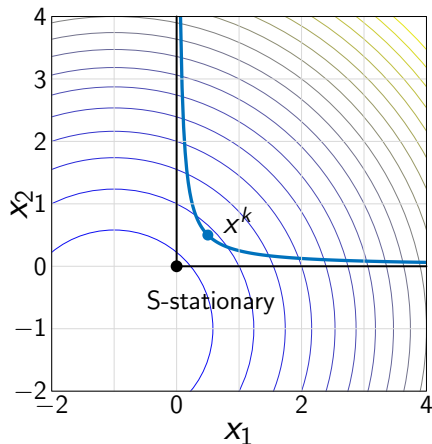
Smoothing method have drawbacks

Usually benefit in using them. Have elaborate convergence theory [Ralph and Wright, 2004].

Consider the two-dimensional MPCC:

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & (x_1 + 1)^2 + (x_2 + 1)^2 \\ \text{s.t.} \quad & 0 \leq x_1 \perp x_2 \geq 0. \end{aligned}$$

- ▶ Relaxation method converges in one iteration to $x^* = (0, 0)$.
- ▶ Smoothing needs an infinite sequence for convergence.



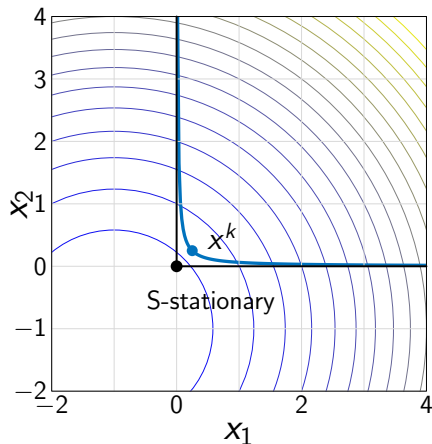
Smoothing method have drawbacks

Usually benefit in using them. Have elaborate convergence theory [Ralph and Wright, 2004].

Consider the two-dimensional MPCC:

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & (x_1 + 1)^2 + (x_2 + 1)^2 \\ \text{s.t.} \quad & 0 \leq x_1 \perp x_2 \geq 0. \end{aligned}$$

- ▶ Relaxation method converges in one iteration to $x^* = (0, 0)$.
- ▶ Smoothing needs an infinite sequence for convergence.



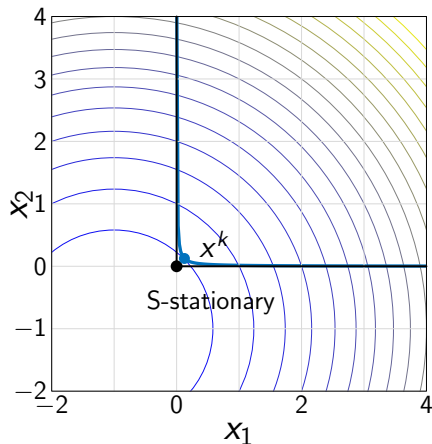
Smoothing method have drawbacks

Usually benefit in using them. Have elaborate convergence theory [Ralph and Wright, 2004].

Consider the two-dimensional MPCC:

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & (x_1 + 1)^2 + (x_2 + 1)^2 \\ \text{s.t.} \quad & 0 \leq x_1 \perp x_2 \geq 0. \end{aligned}$$

- ▶ Relaxation method converges in one iteration to $x^* = (0, 0)$.
- ▶ Smoothing needs an infinite sequence for convergence.



Refresher on interior-point methods

Nonlinear program (NLP)

$$\begin{array}{ll}\min_{x \in \mathbb{R}^n} & f(x) \\ \text{s.t.} & g(x) = 0, \\ & h(x) \geq 0.\end{array}$$

Barrier problem ($\mu > 0$)

$$\begin{array}{ll}\min_{x, s} & f(x) - \mu \sum_i \log s_i \\ \text{s.t.} & g(x) = 0, \\ & h(x) - s = 0, \\ & (s > 0).\end{array}$$

- ▶ Solve a sequence of barrier problems approximately as $\mu \downarrow 0$.
- ▶ Combine Newton steps with a filter line search [Fletcher and Leyffer, 2002] for efficient progress.
- ▶ Regularization and inertia correction provide good Newton steps when the KKT system is singular or has the wrong inertia, for example if LICQ fails or the Lagrangian Hessian is indefinite [Wächter and Biegler, 2006].

Joint σ - μ homotopy for faster solves

Idea due to [Raghunathan and Biegler, 2005] (IPOPT-C), recently revisited in CC0pt [Pozharskiy et al., 2026].

Scholtes relaxation

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) \geq 0, \\ & x_1 \geq 0, \\ & x_2 \geq 0, \\ & x_{1,i}x_{2,i} \leq \sigma^k, \quad i = 1, \dots, m. \end{aligned}$$

Barrier subproblem

$$\begin{aligned} \min_{x,s} \quad & f(x) - \mu^k \left(\sum_j \log s_j \right. \\ & \left. + \sum_{i=1}^m [\log x_{1,i} + \log x_{2,i} \right. \\ & \quad \left. + \log(\sigma^k - x_{1,i}x_{2,i})] \right) \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) - s = 0. \end{aligned}$$

- **Update σ^k and μ^k jointly.** No need to solve each fixed- σ^k NLP to convergence.

Joint σ - μ homotopy for faster solves

Idea due to [Raghunathan and Biegler, 2005] (IPOPT-C), recently revisited in CC0pt [Pozharskiy et al., 2026].

Scholtes relaxation

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) \geq 0, \\ & x_1 \geq 0, \\ & x_2 \geq 0, \\ & x_{1,i} x_{2,i} \leq \sigma^k, \quad i = 1, \dots, m. \end{aligned}$$

Barrier subproblem

$$\begin{aligned} \min_{x,s} \quad & f(x) - \mu^k \left(\sum_j \log s_j \right. \\ & \left. + \sum_{i=1}^m [\log x_{1,i} + \log x_{2,i} \right. \\ & \quad \left. + \log(\sigma^k - x_{1,i} x_{2,i})] \right) \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) - s = 0. \end{aligned}$$

- **Update σ^k and μ^k jointly.** No need to solve each fixed- σ^k NLP to convergence.
- Careful parameter updates matter in practice, both monotone and nonmonotone rules are possible.

Joint σ - μ homotopy for faster solves

Idea due to [Raghunathan and Biegler, 2005] (IPOPT-C), recently revisited in CC0pt [Pozharskiy et al., 2026].

Scholtes relaxation

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) \geq 0, \\ & x_1 \geq 0, \\ & x_2 \geq 0, \\ & x_{1,i} x_{2,i} \leq \sigma^k, \quad i = 1, \dots, m. \end{aligned}$$

Barrier subproblem

$$\begin{aligned} \min_{x,s} \quad & f(x) - \mu^k \left(\sum_j \log s_j \right. \\ & \left. + \sum_{i=1}^m [\log x_{1,i} + \log x_{2,i} \right. \\ & \quad \left. + \log(\sigma^k - x_{1,i} x_{2,i})] \right) \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) - s = 0. \end{aligned}$$

- **Update σ^k and μ^k jointly.** No need to solve each fixed- σ^k NLP to convergence.
- Careful parameter updates matter in practice, both monotone and nonmonotone rules are possible.
- Tailored regularization can be built into the interior-point solver.

Joint σ - μ homotopy for faster solves

Idea due to [Ragunathan and Biegler, 2005] (IPOPT-C), recently revisited in CC0pt [Pozharskiy et al., 2026].

Scholtes relaxation

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) \geq 0, \\ & x_1 \geq 0, \\ & x_2 \geq 0, \\ & x_{1,i}x_{2,i} \leq \sigma^k, \quad i = 1, \dots, m. \end{aligned}$$

Barrier subproblem

$$\begin{aligned} \min_{x,s} \quad & f(x) - \mu^k \left(\sum_j \log s_j \right. \\ & \left. + \sum_{i=1}^m [\log x_{1,i} + \log x_{2,i} \right. \\ & \quad \left. + \log(\sigma^k - x_{1,i}x_{2,i})] \right) \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) - s = 0. \end{aligned}$$

- **Update σ^k and μ^k jointly.** No need to solve each fixed- σ^k NLP to convergence.
- Careful parameter updates matter in practice, both monotone and nonmonotone rules are possible.
- Tailored regularization can be built into the interior-point solver.
- Additional structure in QPCCs can be exploited in the Hessian regularization.

Joint σ - μ homotopy for faster solves

Idea due to [Ragunathan and Biegler, 2005] (IPOPT-C), recently revisited in CC0pt [Pozharskiy et al., 2026].

Scholtes relaxation

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) \geq 0, \\ & x_1 \geq 0, \\ & x_2 \geq 0, \\ & x_{1,i} x_{2,i} \leq \sigma^k, \quad i = 1, \dots, m. \end{aligned}$$

Barrier subproblem

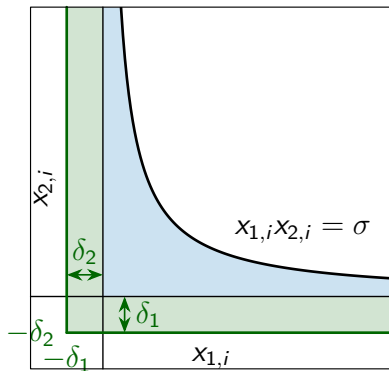
$$\begin{aligned} \min_{x,s} \quad & f(x) - \mu^k \left(\sum_j \log s_j \right. \\ & \left. + \sum_{i=1}^m [\log x_{1,i} + \log x_{2,i} \right. \\ & \quad \left. + \log(\sigma^k - x_{1,i} x_{2,i})] \right) \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) - s = 0. \end{aligned}$$

- **Update σ^k and μ^k jointly.** No need to solve each fixed- σ^k NLP to convergence.
- Careful parameter updates matter in practice, both monotone and nonmonotone rules are possible.
- Tailored regularization can be built into the interior-point solver.
- Additional structure in QPCCs can be exploited in the Hessian regularization.
- You will use CC0pt in the exercises!

Interior-point methods with a two-sided relaxation

An idea from [DeMiguel et al., 2005].

- MPCC feasible sets have no strict interior.



An idea from [DeMiguel et al., 2005].

- $$x_{1,i}x_{2,i} \leq \sigma, \quad i = 1, \dots, m.$$

An idea from [DeMiguel et al., 2005].

- $$x_{1,i}x_{2,i} \leq \sigma, \quad i = 1, \dots, m.$$

- ▶ Relax both sides:

Interior-point methods with a two-sided relaxation

An idea from [DeMiguel et al., 2005].

- ▶ MPCC feasible sets have no strict interior.
- ▶ Scholtes' relaxation

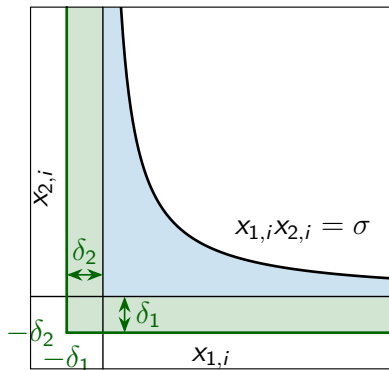
$$x_{1,i}x_{2,i} \leq \sigma, \quad i = 1, \dots, m.$$

has a nonempty interior for $\sigma > 0$, but it collapses as $\sigma \downarrow 0$.

- ▶ Relax both sides:

$$x_{1,i}x_{2,i} \leq \sigma, \quad x_{1,i} \geq -\delta_1, x_{2,i} \geq -\delta_2 \quad i = 1, \dots, m.$$

- ▶ Adjust σ and δ as the barrier parameter $\mu \downarrow 0$.



Interior-point methods with a two-sided relaxation

An idea from [DeMiguel et al., 2005].

- ▶ MPCC feasible sets have no strict interior.
- ▶ Scholtes' relaxation

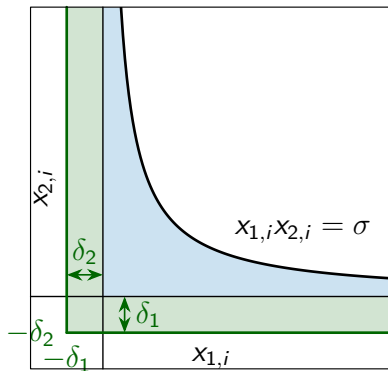
$$x_{1,i}x_{2,i} \leq \sigma, \quad i = 1, \dots, m.$$

has a nonempty interior for $\sigma > 0$, but it collapses as $\sigma \downarrow 0$.

- ▶ Relax both sides:

$$x_{1,i}x_{2,i} \leq \sigma, \quad x_{1,i} \geq -\delta_1, x_{2,i} \geq -\delta_2 \quad i = 1, \dots, m.$$

- ▶ Adjust σ and δ as the barrier parameter $\mu \downarrow 0$.



Some practical takes

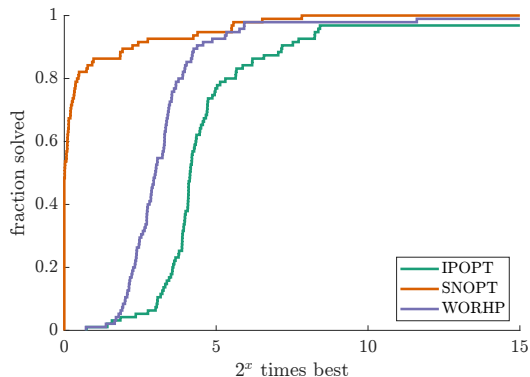
- ▶ If, in the limit, $\sigma \downarrow 0$ or $\delta \downarrow 0$, but not both, the relaxed problem retains a nonempty interior. This can make interior-point methods faster and more robust.
- ▶ In practice, it is difficult to tune when to change $\sigma, \delta_1, \delta_2$ (like detecting active sets).
- ▶ Some advanced heuristics implemented in CCOpt [Pozharskiy et al., 2026]

The benefit of tailored implementations – MacMPEC [Leyffer, 2000]

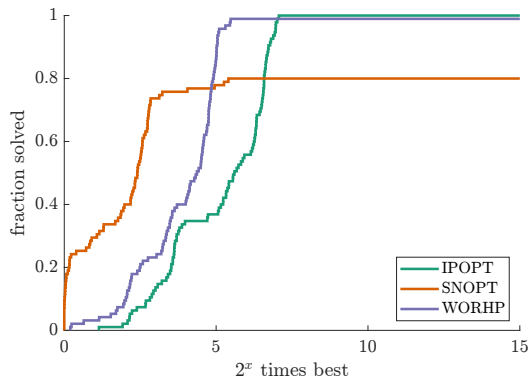
A test suite of MPCCs

- ▶ MacMPEC contains small to medium-sized problems.
- ▶ The benchmark is generally relatively easy to solve.
- ▶ Even direct NLP reformulations solve all feasible problems [Nurkanović et al., 2024].

The benefit of tailored implementations – MacMPEC [Leyffer, 2000]



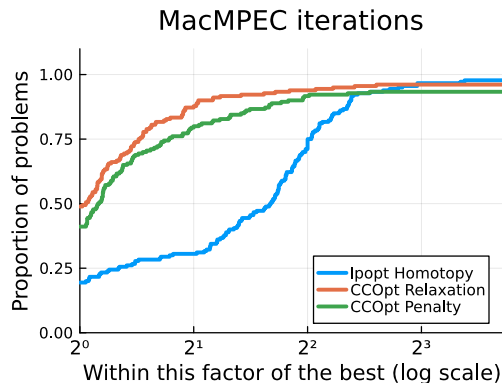
Relative performance on MacMPEC
(solve the MPCC as NLP)



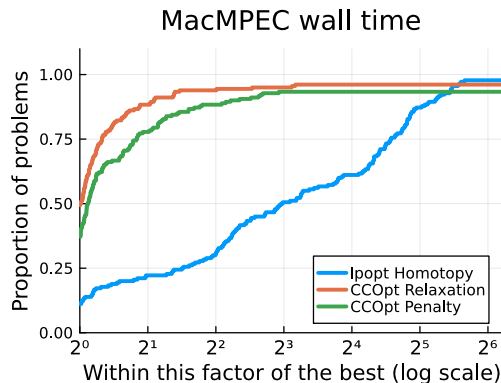
Relative performance on MacMPEC
(Scholtes' homotopy)

A comparison of solvers on the MacMPEC benchmark.

The benefit of tailored implementations – MacMPEC [Leyffer, 2000]



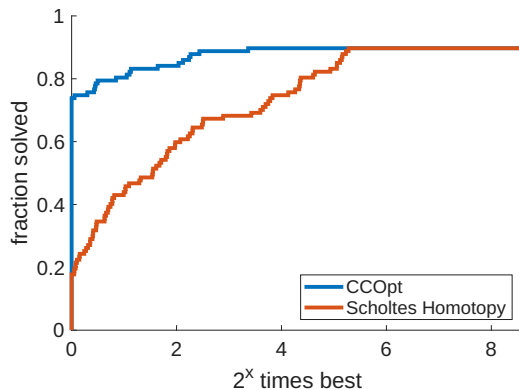
Relative performance on MacMPEC (iterations)



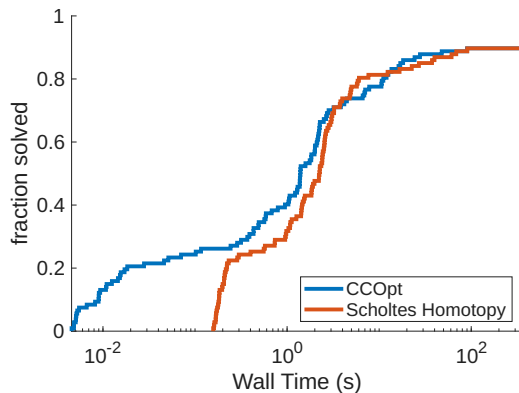
Relative performance on MacMPEC (wall time)

A comparison of solvers on the MacMPEC benchmark.

The NOSBENCH benchmark [N. et al, 2024]



Relative performance on NOSBENCH (wall time)

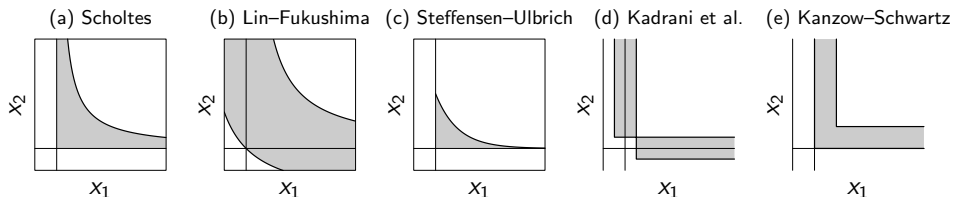


Absolute performance on NOSBENCH (wall time)

A comparison of solvers on the NOSBENCH benchmark.

Other regularization methods

There exist many elaborate ways to relax the L-shaped set. Convergence theory in [Hoheisel et al., 2013]

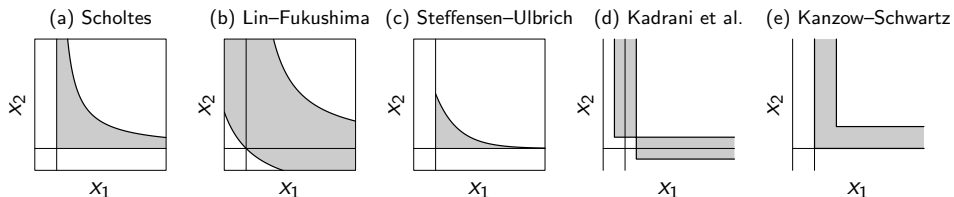


- ▶ They have better convergence properties than Scholtes' method if the NLP's are solved exactly.
- ▶ In practice, they perform better only on easier problems [Nurkanović et al., 2024].

:

Other regularization methods

There exist many elaborate ways to relax the L-shaped set. Convergence theory in [Hoheisel et al., 2013]



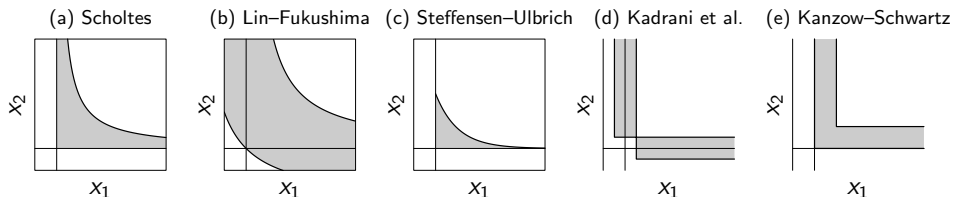
- ▶ They have better convergence properties than Scholtes' method if the NLP's are solved exactly.
- ▶ In practice, they perform better only on easier problems [Nurkanović et al., 2024].

Convergence results [Hoheisel et al., 2013]:

- a) [Scholtes, 2001]: C-stationarity under MPCC-MFCQ.

Other regularization methods

There exist many elaborate ways to relax the L-shaped set. Convergence theory in [Hoheisel et al., 2013]



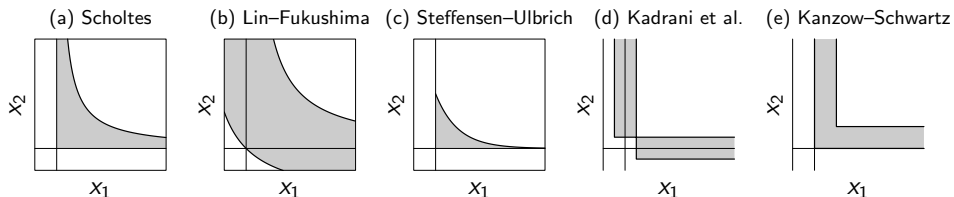
- ▶ They have better convergence properties than Scholtes' method if the NLP's are solved exactly.
- ▶ In practice, they perform better only on easier problems [Nurkanović et al., 2024].

Convergence results [Hoheisel et al., 2013]:

- a) [Scholtes, 2001]: C-stationarity under MPCC-MFCQ.
- b) [Lin and Fukushima, 2005]: C-stationarity under MPCC-MFCQ.

Other regularization methods

There exist many elaborate ways to relax the L-shaped set. Convergence theory in [Hoheisel et al., 2013]



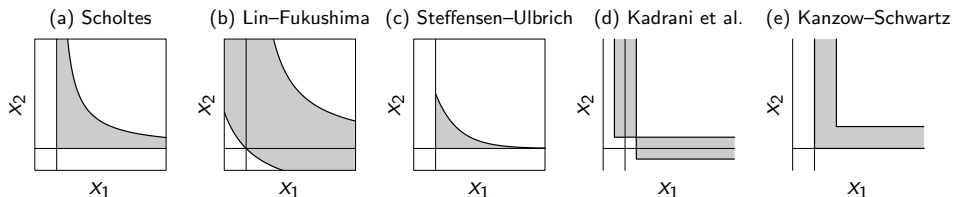
- ▶ They have better convergence properties than Scholtes' method if the NLP's are solved exactly.
- ▶ In practice, they perform better only on easier problems [Nurkanović et al., 2024].

Convergence results [Hoheisel et al., 2013]:

- a) [Scholtes, 2001]: C-stationarity under MPCC-MFCQ.
- b) [Lin and Fukushima, 2005]: C-stationarity under MPCC-MFCQ.
- c) [Steffensen and Ulbrich, 2010]: C-stationarity under MPCC-CPLD.

Other regularization methods

There exist many elaborate ways to relax the L-shaped set. Convergence theory in [Hoheisel et al., 2013]



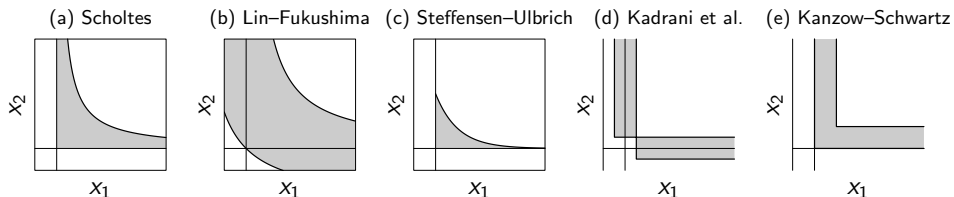
- ▶ They have better convergence properties than Scholtes' method if the NLP's are solved exactly.
- ▶ In practice, they perform better only on easier problems [Nurkanović et al., 2024].

Convergence results [Hoheisel et al., 2013]:

- a) [Scholtes, 2001]: C-stationarity under MPCC-MFCQ.
- b) [Lin and Fukushima, 2005]: C-stationarity under MPCC-MFCQ.
- c) [Steffensen and Ulbrich, 2010]: C-stationarity under MPCC-CPLD.
- d) [Kadrani et al., 2009]: **M-stationarity** under MPCC-CPLD.

Other regularization methods

There exist many elaborate ways to relax the L-shaped set. Convergence theory in [Hoheisel et al., 2013]



- ▶ They have better convergence properties than Scholtes' method if the NLP's are solved exactly.
- ▶ In practice, they perform better only on easier problems [Nurkanović et al., 2024].

Convergence results [Hoheisel et al., 2013]:

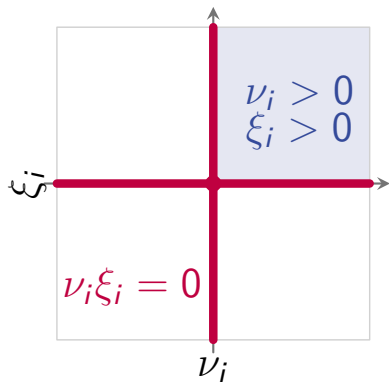
- a) [Scholtes, 2001]: C-stationarity under MPCC-MFCQ.
- b) [Lin and Fukushima, 2005]: C-stationarity under MPCC-MFCQ.
- c) [Steffensen and Ulbrich, 2010]: C-stationarity under MPCC-CPLD.
- d) [Kadrani et al., 2009]: **M-stationarity** under MPCC-CPLD.
- e) [Kanzow and Schwartz, 2013]: **M-stationarity** under MPCC-CPLD.

The problem of M-stationarity in finite precision

A ghost that you never catch

Exact M-stationarity

$$(\nu_i > 0, \xi_i > 0) \quad \text{or} \quad \nu_i \xi_i = 0$$

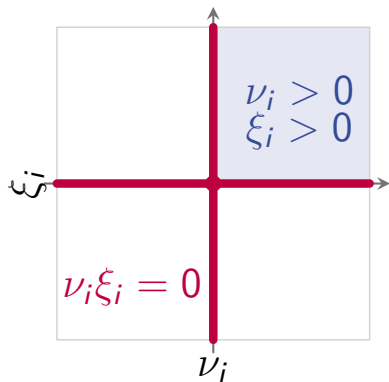


The problem of M-stationarity in finite precision

A ghost that you never catch

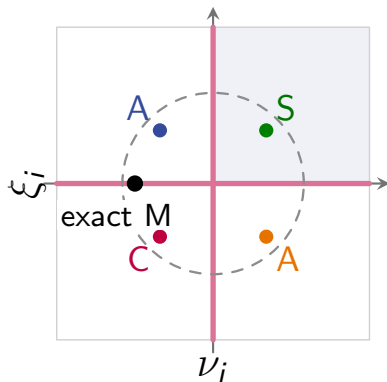
Exact M-stationarity

$$(\nu_i > 0, \xi_i > 0) \quad \text{or} \quad \nu_i \xi_i = 0$$



Finite-precision perturbations

$$(\hat{\nu}_i, \hat{\xi}_i) = (\nu_i, \xi_i) + \mathcal{O}(\varepsilon)$$

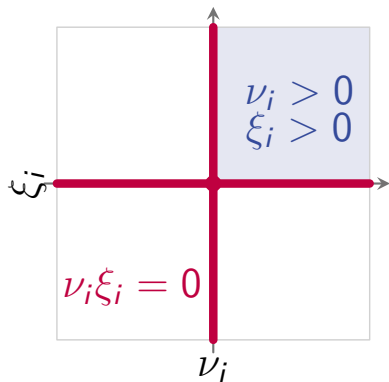


The problem of M-stationarity in finite precision

A ghost that you never catch

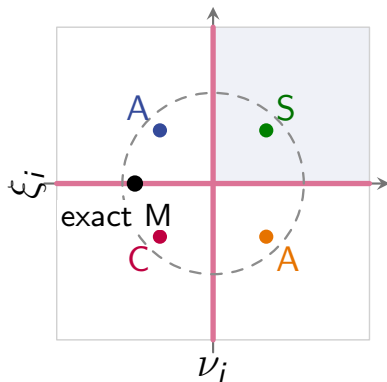
Exact M-stationarity

$$(\nu_i > 0, \xi_i > 0) \quad \text{or} \quad \nu_i \xi_i = 0$$



Finite-precision perturbations

$$(\hat{\nu}_i, \hat{\xi}_i) = (\nu_i, \xi_i) + \mathcal{O}(\varepsilon)$$



Finite-precision issue

An M-stationary multiplier pair is either S-stationary or lies exactly on an axis. A perturbation off the axis therefore either enters the S-region or destroys M-stationarity.

“The price of inexactness” [Kanzow and Schwartz, 2015]

- ▶ Most convergence results derived under the assumption that every subproblem for a fixed σ^k is solved exactly.
- ▶ If NLPs are solved to some $\epsilon > 0$ threshold, than most methods have weaker convergence properties.

“The price of inexactness” [Kanzow and Schwartz, 2015]

- ▶ Most convergence results derived under the assumption that every subproblem for a fixed σ^k is solved exactly.
- ▶ If NLPs are solved to some $\epsilon > 0$ threshold, than most methods have weaker convergence properties.
- ▶ Steffensen-Ulbrich, Kadrani, Kanzow-Schwartz converge only to **W-stationary** points!

“The price of inexactness” [Kanzow and Schwartz, 2015]

- ▶ Most convergence results derived under the assumption that every subproblem for a fixed σ^k is solved exactly.
- ▶ If NLPs are solved to some $\epsilon > 0$ threshold, than most methods have weaker convergence properties.
- ▶ Steffensen-Ulbrich, Kadrani, Kanzow-Schwartz converge only to **W-stationary** points!
- ▶ Scholtes' method still converges to C-stationary points!

Penalty methods

Loosely related to regularization methods [Leyffer et al., 2006]

Main idea: put difficult part into objective.

The ℓ_1 reformulation

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) + \rho x_1^\top x_2 \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) \geq 0, \\ & x_1, x_2 \geq 0. \end{aligned}$$

The ℓ_∞ reformulation

$$\begin{aligned} \min_{x \in \mathbb{R}^n, s \in \mathbb{R}} \quad & f(x) + \rho s \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) \geq 0, \\ & x_1, x_2 \geq 0, \\ & x_{1,i} x_{2,i} \leq s, \quad i = 1 \dots m, \\ & 0 \leq s \leq \bar{s}. \end{aligned}$$

Penalty methods

Loosely related to regularization methods [Leyffer et al., 2006]

Main idea: put difficult part into objective.

The ℓ_1 reformulation

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) + \rho x_1^\top x_2 \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) \geq 0, \\ & x_1, x_2 \geq 0. \end{aligned}$$

The ℓ_∞ reformulation

$$\begin{aligned} \min_{x \in \mathbb{R}^n, s \in \mathbb{R}} \quad & f(x) + \rho s \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) \geq 0, \\ & x_1, x_2 \geq 0, \\ & x_{1,i} x_{2,i} \leq s, \quad i = 1 \dots m, \\ & 0 \leq s \leq \bar{s}. \end{aligned}$$

- ▶ For sufficiently large penalty parameter ρ solve single NLP to solve MPCC [Anitescu, 2005a, Anitescu, 2005b, Ralph and Wright, 2004].
- ▶ Good implementations update the penalty parameter in a homotopy $\rho^k = \frac{1}{\sigma^k}$
- ▶ In practice, usually converges faster than Scholtes' method, but has a lower success rate.

Steering the homotopy parameter to zero

The practical performance depends on the update rate.

Approach: Solve a sequence of regularized NLPs (σ^k) , warm start the next iteration with $x^*(\sigma^k)$. Update the homotopy parameter via:

- Linear update rule:

$$\sigma^{k+1} = \kappa \sigma^k, \quad \kappa \in (0, 1)$$

- For faster convergence, superlinear rules may also be helpful:

$$\sigma^{k+1} = \min(\kappa \sigma^k, (\sigma^k)^\eta), \quad \kappa \in (0, 1), \quad \eta > 1$$

Steering the homotopy parameter to zero

The practical performance depends on the update rate.

Approach: Solve a sequence of regularized NLPs (σ^k) , warm start the next iteration with $x^*(\sigma^k)$. Update the homotopy parameter via:

- ▶ Linear update rule:

$$\sigma^{k+1} = \kappa \sigma^k, \quad \kappa \in (0, 1)$$

- ▶ For faster convergence, superlinear rules may also be helpful:

$$\sigma^{k+1} = \min(\kappa \sigma^k, (\sigma^k)^\eta), \quad \kappa \in (0, 1), \quad \eta > 1$$

Some best practices:

- ▶ Start with not too large, and not too small σ^0 . For well-scaled problems $\sigma^0 = 1$ usually works well.
- ▶ Do not update too aggressively nor too conservatively. Good choices are $\kappa = 0.1$, $\kappa = 0.2$ and $\eta = 1.5$.
- ▶ Update rules are usually monotone. To increase robustness, if an iteration fails, go back and update with $\kappa^+ = \gamma \kappa$, e.g., $\gamma = 2$.

Practical recommendations

- ▶ For tiny problems, almost any method will work.
- ▶ For small- to medium-scale problems, SQP-based methods applied directly or within a homotopy are often faster than interior-point methods.

Practical recommendations

- ▶ For tiny problems, almost any method will work.
- ▶ For small- to medium-scale problems, SQP-based methods applied directly or within a homotopy are often faster than interior-point methods.

Practical recommendations

- ▶ For tiny problems, almost any method will work.
- ▶ For small- to medium-scale problems, SQP-based methods applied directly or within a homotopy are often faster than interior-point methods.
- ▶ A simple and usually efficient approach is to combine your preferred NLP solver with the Scholtes relaxation.
- ▶ Choose the update of $\sigma^k \rightarrow 0$ carefully: neither too slowly nor too aggressively.
- ▶ More elaborate regularization methods are rarely worth the implementation effort.

Practical recommendations

- ▶ For tiny problems, almost any method will work.
- ▶ For small- to medium-scale problems, SQP-based methods applied directly or within a homotopy are often faster than interior-point methods.
- ▶ A simple and usually efficient approach is to combine your preferred NLP solver with the Scholtes relaxation.
- ▶ Choose the update of $\sigma^k \rightarrow 0$ carefully: neither too slowly nor too aggressively.
- ▶ More elaborate regularization methods are rarely worth the implementation effort.
- ▶ Generic homotopy methods are available through `mpccsol` in `nosnoc` for Python and MATLAB.

Practical recommendations

- ▶ For tiny problems, almost any method will work.
- ▶ For small- to medium-scale problems, SQP-based methods applied directly or within a homotopy are often faster than interior-point methods.
- ▶ A simple and usually efficient approach is to combine your preferred NLP solver with the Scholtes relaxation.
- ▶ Choose the update of $\sigma^k \rightarrow 0$ carefully: neither too slowly nor too aggressively.
- ▶ More elaborate regularization methods are rarely worth the implementation effort.
- ▶ Generic homotopy methods are available through `mpccsol` in `nosnoc` for Python and MATLAB.
- ▶ Since CasADi 3.8.0, `CCOpt` can be used directly and is typically faster and more robust than basic homotopy implementations.

Practical recommendations

- ▶ For tiny problems, almost any method will work.
- ▶ For small- to medium-scale problems, SQP-based methods applied directly or within a homotopy are often faster than interior-point methods.
- ▶ A simple and usually efficient approach is to combine your preferred NLP solver with the Scholtes relaxation.
- ▶ Choose the update of $\sigma^k \rightarrow 0$ carefully: neither too slowly nor too aggressively.
- ▶ More elaborate regularization methods are rarely worth the implementation effort.
- ▶ Generic homotopy methods are available through `mpccsol` in `nosnoc` for Python and MATLAB.
- ▶ Since CasADi 3.8.0, `CCOpt` can be used directly and is typically faster and more robust than basic homotopy implementations.

Try `CCOpt` and `mpccsol`:

 MPCC examples in 

 MPCC examples in 

Outline of the lecture

1. Overview of MPCC methods

2. Regularization and penalty methods

Regularization methods

Penalty methods

3. Combinatorial and active-set methods

A sequential linearization method for bound-constrained MPCCs

A sequential LPCC-EQP method

MPECopt: an algorithm for computing a B-stationary points

Sequential quadratic programming with complementarity constraints

Motivation for active set methods for MPCCs

Difficulties:

1. Regularization methods, also under very strong assumptions, converge to points that are weaker than S-stationary (which are possibly not B-stationary).
2. Regularized NLPs, small or large σ , may be difficult to solve (more than TNLPs/BNLPs).

Motivation for active set methods for MPCCs

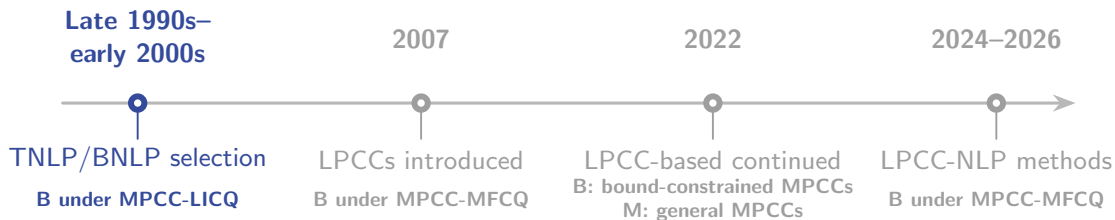
Difficulties:

1. Regularization methods, also under very strong assumptions, converge to points that are weaker than S-stationary (which are possibly not B-stationary).
2. Regularized NLPs, small or large σ , may be difficult to solve (more than TNLPs/BNLPs).

Main idea of active-set methods:

1. Find a feasible point $x \in \Omega$ of the MPCC (1).
2. (Approximately) solve the branch NLP with some partition of the index set, e.g. $\mathcal{I}_1(x)$, $\mathcal{I}_2(x)$
3. Call an oracle to verify B-stationary or to give a better $\mathcal{I}_1(x^+)$ and $\mathcal{I}_2(x^+)$ (e.g. an LPCC)
4. If not B-stationary, go to 2.

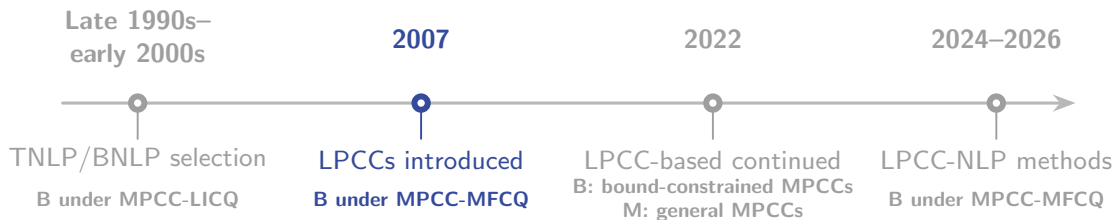
A brief history of active-set/combinatorial/pivoting methods



Multiplier-based TNLP/BNLP selection. The next TNLP/BNLP is selected from the multiplier signs. B-stationarity can be guaranteed only under MPCC-LICQ.

[Fukushima and Tseng, 2002, Giallombardo and Ralph, 2008, Izmailov and Solodov, 2008, Jiang and Ralph, 1999, Lin and Fukushima, 2006, Liu et al., 2006, Luo et al., 1996, Scholtes and Stöhr, 1999]

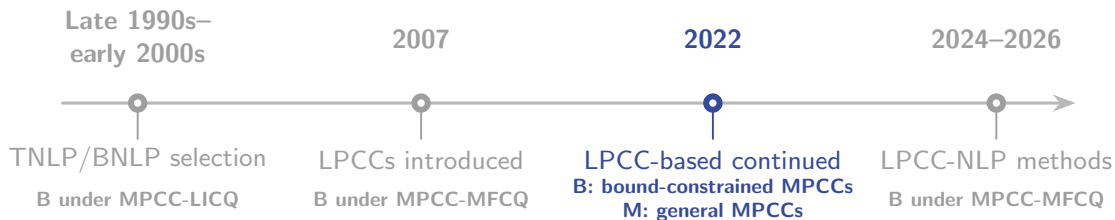
A brief history of active-set/combinatorial/pivoting methods



The LPCC introduced. LPCCs are introduced both as stopping tests and as subproblems for computing steps, yielding B-stationarity under MPCC-MFCQ.

[Leyffer and Munson, 2007]

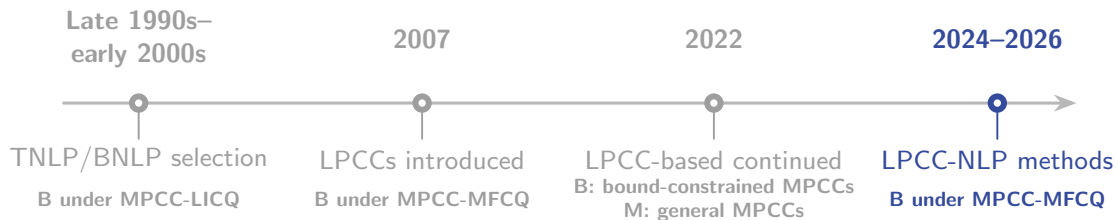
A brief history of active-set/combinatorial/pivoting methods



LPCC-based methods continued. For bound-constrained MPCCs they guarantee B-stationarity. LPCC solvable in linear runtime. An augmented-Lagrangian extension for general MPCCs guarantees M-stationarity.

[Kirches et al., 2022, Guo and Deng, 2022]

A brief history of active-set/combinatorial/pivoting methods



General MPCCs. LPCCs combined with std. NLP methods achieve B-stationarity under MPCC-MFCQ.

[Kazi et al., 2024, Nurkanović and Leyffer, 2026]

Refresher on the Branch Nonlinear Program (BNLP)

MPCC active sets

$$\mathcal{I}_{+0}(x) = \{i \in \{1, \dots, m\} \mid x_{1,i} > 0, x_{2,i} = 0\},$$

$$\mathcal{I}_{0+}(x) = \{i \in \{1, \dots, m\} \mid x_{1,i} = 0, x_{2,i} > 0\},$$

$$\mathcal{I}_{00}(x) = \{i \in \{1, \dots, m\} \mid x_{1,i} = 0, x_{2,i} = 0\}.$$

Partitioning of $\mathcal{I}_{00}(x)$

$$\mathcal{D}_1(x) \cup \mathcal{D}_2(x) = \mathcal{I}_{00}(x),$$

$$\mathcal{D}_1(x) \cap \mathcal{D}_2(x) = \emptyset,$$

$$\mathcal{I}_1(x) := \mathcal{I}_{0+}(x) \cup \mathcal{D}_1(x),$$

$$\mathcal{I}_2(x) := \mathcal{I}_{+0}(x) \cup \mathcal{D}_2(x),$$

Branch NLP (BNLP) at x^*

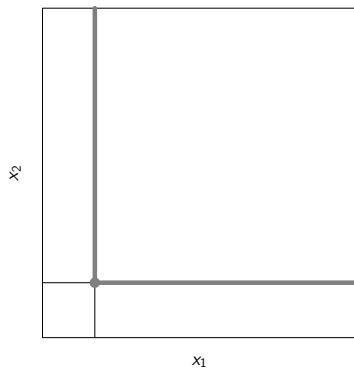
$$\min_{x \in \mathbb{R}^n} f(x)$$

$$\text{s.t.} \quad g(x) = 0,$$

$$h(x) \geq 0,$$

$$x_{1,i} = 0, \quad x_{2,i} \geq 0, \quad i \in \mathcal{I}_1(x^*),$$

$$x_{1,i} \geq 0, \quad x_{2,i} = 0, \quad i \in \mathcal{I}_2(x^*).$$



Refresher on the Branch Nonlinear Program (BNLP)

MPCC active sets

$$\mathcal{I}_{+0}(x) = \{i \in \{1, \dots, m\} \mid x_{1,i} > 0, x_{2,i} = 0\},$$

$$\mathcal{I}_{0+}(x) = \{i \in \{1, \dots, m\} \mid x_{1,i} = 0, x_{2,i} > 0\},$$

$$\mathcal{I}_{00}(x) = \{i \in \{1, \dots, m\} \mid x_{1,i} = 0, x_{2,i} = 0\}.$$

Partitioning of $\mathcal{I}_{00}(x)$

$$\mathcal{D}_1(x) \cup \mathcal{D}_2(x) = \mathcal{I}_{00}(x),$$

$$\mathcal{D}_1(x) \cap \mathcal{D}_2(x) = \emptyset,$$

$$\mathcal{I}_1(x) := \mathcal{I}_{0+}(x) \cup \mathcal{D}_1(x),$$

$$\mathcal{I}_2(x) := \mathcal{I}_{+0}(x) \cup \mathcal{D}_2(x),$$

Branch NLP (BNLP) at x^*

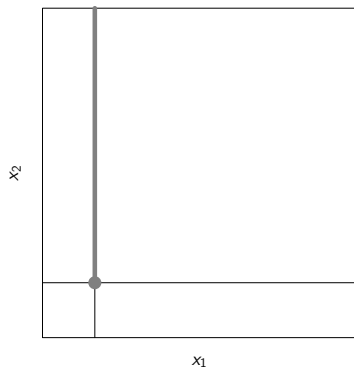
$$\min_{x \in \mathbb{R}^n} f(x)$$

$$\text{s.t.} \quad g(x) = 0,$$

$$h(x) \geq 0,$$

$$x_{1,i} = 0, \quad x_{2,i} \geq 0, \quad i \in \mathcal{I}_1(x^*),$$

$$x_{1,i} \geq 0, \quad x_{2,i} = 0, \quad i \in \mathcal{I}_2(x^*).$$



Refresher on the Branch Nonlinear Program (BNLP)

MPCC active sets

$$\mathcal{I}_{+0}(x) = \{i \in \{1, \dots, m\} \mid x_{1,i} > 0, x_{2,i} = 0\},$$

$$\mathcal{I}_{0+}(x) = \{i \in \{1, \dots, m\} \mid x_{1,i} = 0, x_{2,i} > 0\},$$

$$\mathcal{I}_{00}(x) = \{i \in \{1, \dots, m\} \mid x_{1,i} = 0, x_{2,i} = 0\}.$$

Partitioning of $\mathcal{I}_{00}(x)$

$$\mathcal{D}_1(x) \cup \mathcal{D}_2(x) = \mathcal{I}_{00}(x),$$

$$\mathcal{D}_1(x) \cap \mathcal{D}_2(x) = \emptyset,$$

$$\mathcal{I}_1(x) := \mathcal{I}_{0+}(x) \cup \mathcal{D}_1(x),$$

$$\mathcal{I}_2(x) := \mathcal{I}_{+0}(x) \cup \mathcal{D}_2(x),$$

Branch NLP (BNLP) at x^*

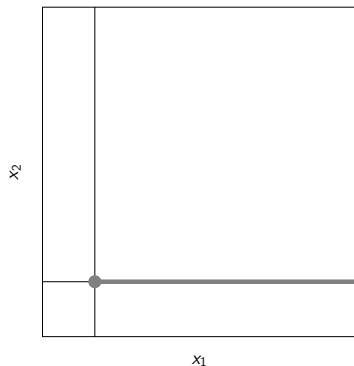
$$\min_{x \in \mathbb{R}^n} f(x)$$

$$\text{s.t.} \quad g(x) = 0,$$

$$h(x) \geq 0,$$

$$x_{1,i} = 0, \quad x_{2,i} \geq 0, \quad i \in \mathcal{I}_1(x^*),$$

$$x_{1,i} \geq 0, \quad x_{2,i} = 0, \quad i \in \mathcal{I}_2(x^*).$$



The B-stationarity concept via LPCCs

If there are no complementarity constraints, the LPCC reduces to an LP.

Theorem ([Luo et al., 1996, Scholtes, 2004, Flegel and Kanzow, 2005])

Let $\mathcal{T}_\Omega(x^k) = \mathcal{T}_\Omega^{\text{MPCC}}(x^k)$, and $x^k \in \Omega$ be a local minimizer of the MPCC (1), then it holds that

$$\nabla f(x^k)^\top d \geq 0 \text{ for all } d \in \mathcal{T}_\Omega^{\text{MPCC}}(x^k), \quad (2)$$

or equivalently, $d = 0$ is a global minimizer of the following optimization problem:

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \nabla f(x^k)^\top d \\ \text{s.t.} \quad & g(x^k) + \nabla g(x^k)^\top d = 0, \\ & h(x^k) + \nabla h(x^k)^\top d \geq 0, i \in \mathcal{A}^k \\ & x_{1,i}^k + d_{1,i} = 0, \forall i \in \mathcal{I}_{0+}^k, \\ & x_{2,i}^k + d_{2,i} = 0, \forall i \in \mathcal{I}_{+0}^k, \\ & 0 \leq x_{1,i}^k + d_{1,i} \perp x_{2,i}^k + d_{2,i} \geq 0, \forall i \in \mathcal{I}_{00}^k \quad (\text{degenerate}) \end{aligned}$$

B-stationarity via reduced and full LPCC

Use active sets at x^k , reduced LPCC.

Definition (B-stationarity via reduced [Flegel and Kanzow, 2005])

A feasible point x^k is called a B-stationary point if $d = 0$ is the global minimizer of the following LPCC:

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \nabla f(x^k)^\top d \\ \text{s.t.} \quad & g(x^k) + \nabla g(x^k)^\top d = 0, \\ & h_i(x^k) + \nabla h_i(x^k)^\top d \geq 0, \quad \forall i \in \mathcal{A}^k \\ & x_{1,i}^k + d_{1,i} = 0, \quad x_{2,i}^k + d_{2,i} \geq 0, \quad \forall i \in \mathcal{I}_{0+}, \\ & x_{1,i}^k + d_{1,i} \geq 0, \quad x_{2,i}^k + d_{2,i} = 0, \quad \forall i \in \mathcal{I}_{+0}, \\ & 0 \leq x_{1,i}^k + d_{1,i} \perp x_{2,i}^k + d_{2,i} \geq 0, \quad \forall i \in \mathcal{I}_{00}, \\ & (\|d\|_\infty \leq \rho). \end{aligned}$$

B-stationarity via reduced and full LPCC

Use active sets at x^k , reduced LPCC.

Definition (B-stationarity via reduced [Flegel and Kanzow, 2005])

A feasible point x^k is called a B-stationary point if $d = 0$ is the global minimizer of the following LPCC:

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \nabla f(x^k)^\top d \\ \text{s.t.} \quad & g(x^k) + \nabla g(x^k)^\top d = 0, \\ & h_i(x^k) + \nabla h_i(x^k)^\top d \geq 0, \forall i \in \mathcal{A}^k \\ & x_{1,i}^k + d_{1,i} = 0, \quad x_{2,i}^k + d_{2,i} \geq 0, \forall i \in \mathcal{I}_{0+}, \\ & x_{1,i}^k + d_{1,i} \geq 0, \quad x_{2,i}^k + d_{2,i} = 0, \forall i \in \mathcal{I}_{+0}, \\ & 0 \leq x_{1,i}^k + d_{1,i} \perp x_{2,i}^k + d_{2,i} \geq 0, \forall i \in \mathcal{I}_{00}^k, \\ & (\|d\|_\infty \leq \rho). \end{aligned}$$

Use all complementarities, full LPCC.

Definition (B-stationarity via full [Luo et al., 1996])

A feasible point x^k is called a B-stationary point if $d = 0$ is a minimizer the following LPCC:

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \nabla f(x^k)^\top d \\ \text{s.t.} \quad & g(x^k) + \nabla g(x^k)^\top d = 0, \\ & h(x^k) + \nabla h(x^k)^\top d \geq 0, \\ & 0 \leq x_{1,i}^k + d_{1,i} \perp x_{2,i}^k + d_{2,i} \geq 0, \forall i, \\ & (\|d\|_\infty \leq \rho). \end{aligned}$$

Full vs reduced trust-region LPCC

LPCC(x^k, ρ) – full

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \nabla f(x^k)^\top d \\ \text{s.t.} \quad & g(x^k) + \nabla g(x^k)^\top d = 0, \\ & h(x^k) + \nabla h(x^k)^\top d \geq 0, \\ & 0 \leq x_{1,i}^k + d_{1,i} \perp x_{2,i}^k + d_{2,i} \geq 0, \forall i, \\ & \|d\|_\infty \leq \rho, \end{aligned}$$

- ▶ If x^k infeasible, then not clear how to define index sets in reduced LPCC.
- ▶ If x^k not B-stationary, then $\nabla f(x^k)^\top d < 0$.
- ▶ Use $x^k + d$ for next active set guess.

Full vs reduced trust-region LPCC

LPCC(x^k, ρ) – full

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \nabla f(x^k)^\top d \\ \text{s.t.} \quad & g(x^k) + \nabla g(x^k)^\top d = 0, \\ & h(x^k) + \nabla h(x^k)^\top d \geq 0, \\ & 0 \leq x_{1,i}^k + d_{1,i} \perp x_{2,i}^k + d_{2,i} \geq 0, \forall i, \\ & \|d\|_\infty \leq \rho, \end{aligned}$$

- ▶ If x^k infeasible, then not clear how to define index sets in reduced LPCC.
- ▶ If x^k not B-stationary, then $\nabla f(x^k)^\top d < 0$.
- ▶ Use $x^k + d$ for next active set guess.

Lemma

Let $x^k \in \Omega$ be a feasible point of the MPCC (1). For all sufficiently small trust region radii, $\rho > 0$, the sets of the local minimizers of the reduced and full LPCC are identical.

From now on, we only use the *full* LPCC.

Outline of the lecture

1. Overview of MPCC methods

2. Regularization and penalty methods

3. Combinatorial and active-set methods

A sequential linearization method for bound-constrained MPCCs

A sequential LPCC-EQP method

MPECopt: an algorithm for computing a B-stationary points

Sequential quadratic programming with complementarity constraints

From a generic MPCC to bound constraints

Method and aug. Lag. extension introduced in [Kirches et al., 2022], aug. Lag. studied in [Guo and Deng, 2022]

Generic MPCC

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) \geq 0, \\ & 0 \leq x_1 \perp x_2 \geq 0. \end{aligned}$$

Introduce $s \geq 0$ with $h(x) - s = 0$. For fixed multipliers λ, ν and penalty parameter $\beta > 0$, define

$$\mathcal{L}_\beta(x, s; \lambda, \nu) = f(x) - \lambda^\top g(x) - \nu^\top (h(x) - s) + \frac{\beta}{2} (\|g(x)\|_2^2 + \|h(x) - s\|_2^2).$$

The augmented-Lagrangian subproblem is a bound-constrained MPCC:

$$\min_{x, s} \mathcal{L}_\beta(x, s; \lambda, \nu) \quad \text{s.t.} \quad s \geq 0, \quad 0 \leq x_1 \perp x_2 \geq 0.$$

Under suitable assumptions, and penalty parameter updates, a sequence of solutions of this subproblem converges to M-stationary points [Guo and Deng, 2022].

Bound-constrained MPCCs

The special case used by the sequential LPCC method [Kirches et al., 2022].

Bound-constrained MPCC

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & \ell_x \leq x_0 \leq u_x, \\ & 0 \leq x_1 \perp x_2 \geq 0. \end{aligned}$$

$$x = (x_0, x_1, x_2) \in \mathbb{R}^n, \quad x_0 \in \mathbb{R}^{n_0}, \quad x_1, x_2 \in \mathbb{R}^m, \quad n = n_0 + 2m.$$

- ▶ A smooth objective, simple bounds on x_0 , and m complementarity pairs.
- ▶ Subproblem in aug. Lag. methods for MPCCs.

LPCC step and branch QP

The LPCC predicts a branch, the BQP improves the step on that branch.

LPCC(x^k, ρ)

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \nabla f(x^k)^\top d \\ \text{s.t.} \quad & \ell_x \leq x_0^k + d_0 \leq u_x, \\ & 0 \leq x_1^k + d_1 \perp x_2^k + d_2 \geq 0, \\ & \|d\|_\infty \leq \rho. \end{aligned}$$

BQP($x^k, \rho; \mathcal{I}_1, \mathcal{I}_2$)

$$\begin{aligned} \min_{p \in \mathbb{R}^n} \quad & \nabla f(x^k)^\top p + \frac{1}{2} p^\top H^k p \\ \text{s.t.} \quad & \ell_x \leq x_0^k + p_0 \leq u_x, \\ & x_{1,i}^k + p_{1,i} = 0, \quad i \in \mathcal{I}_1, \\ & x_{2,i}^k + p_{2,i} \geq 0, \quad i \in \mathcal{I}_1, \\ & x_{1,i}^k + p_{1,i} \geq 0, \quad i \in \mathcal{I}_2, \\ & x_{2,i}^k + p_{2,i} = 0, \quad i \in \mathcal{I}_2, \\ & \|p\|_\infty \leq \rho. \end{aligned}$$

Let $\bar{x} = x^k + d_L$ be the LPCC trial point. Split its biactive set as

$$\begin{aligned} \mathcal{D}_1 &= \{i \in \mathcal{I}_{00}(\bar{x}) \mid \nabla_{x_{1,i}} f(\bar{x}) \geq \nabla_{x_{2,i}} f(\bar{x})\}, & \mathcal{D}_2 &= \mathcal{I}_{00}(\bar{x}) \setminus \mathcal{D}_1, \\ \mathcal{I}_1 &= \mathcal{I}_{0+}(\bar{x}) \cup \mathcal{D}_1, & \mathcal{I}_2 &= \mathcal{I}_{+0}(\bar{x}) \cup \mathcal{D}_2. \end{aligned}$$

Sequential LPCC algorithm

For the bound-constrained MPCC [Kirches et al., 2022].

Given: feasible x^0 , $\bar{\rho}^0 > \underline{\rho} > 0$, and acceptance threshold $\eta \in (0, 1)$.

For $k = 0, 1, \dots$:

1. Choose an inner trust-region radius $\rho^{k,0} \in [\underline{\rho}, \bar{\rho}^k]$.

2. **For** $\ell = 0, 1, \dots$:

2.1 Solve LPCC($x^k, \rho^{k,\ell}$) for a step $d^{k,\ell}$.

2.2 If $d = 0$ is a local minimizer of the LPCC, stop: x^k is B-stationary.

2.3 Compute the ratio of actual to predicted decrease (“sufficient decrease conditions”):

$$r^{k,\ell} = \frac{f(x^k) - f(x^k + d^{k,\ell})}{-\nabla f(x^k)^\top d^{k,\ell}}.$$

2.4 If $r^{k,\ell} \geq \eta$, accept the step:

$$x^{k+1} = x^k + d^{k,\ell}, \quad \bar{\rho}^{k+1} = \max\{\bar{\rho}^k, 2\rho^{k,\ell}\},$$

and leave the inner loop. Otherwise set $\rho^{k,\ell+1} = \rho^{k,\ell}/2$.

3. Optionally refine the accepted iterate with a BQP($x^k, \rho; \mathcal{I}_1, \mathcal{I}_2$) on the predicted branch.

The LPCC decomposes componentwise

There are no coupling constraint between variables, we can split the the LPCC into small subproblems

LPCC(x^k, ρ)

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \nabla f(x^k)^\top d \\ \text{s.t.} \quad & \ell_x \leq x_0^k + d_0 \leq u_x, \\ & 0 \leq x_1^k + d_1 \perp x_2^k + d_2 \geq 0, \\ & \|d\|_\infty \leq \rho. \end{aligned}$$

One scalar LP for each $x_{0,j}$

$$\begin{aligned} \min_{d_{0,j} \in \mathbb{R}} \quad & (\nabla_{x_0} f(x^k))_j d_{0,j} \\ \text{s.t.} \quad & \ell_{x,j} \leq x_{0,j}^k + d_{0,j} \leq u_{x,j}, \\ & |d_{0,j}| \leq \rho. \end{aligned}$$

One two-dimensional LPCC for each pair i

$$\begin{aligned} \min_{(d_{1,i}, d_{2,i}) \in \mathbb{R}^2} \quad & (\nabla_{x_1} f(x^k))_i d_{1,i} + (\nabla_{x_2} f(x^k))_i d_{2,i} \\ \text{s.t.} \quad & 0 \leq x_{1,i}^k + d_{1,i} \perp x_{2,i}^k + d_{2,i} \geq 0, \\ & |d_{1,i}| \leq \rho, \\ & |d_{2,i}| \leq \rho. \end{aligned}$$

The LPCC decomposes componentwise

There are no coupling constraint between variables, we can split the the LPCC into small subproblems

LPCC(x^k, ρ)

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \nabla f(x^k)^\top d \\ \text{s.t.} \quad & \ell_x \leq x_0^k + d_0 \leq u_x, \\ & 0 \leq x_1^k + d_1 \perp x_2^k + d_2 \geq 0, \\ & \|d\|_\infty \leq \rho. \end{aligned}$$

One scalar LP for each $x_{0,j}$

$$\begin{aligned} \min_{d_{0,j} \in \mathbb{R}} \quad & (\nabla_{x_0} f(x^k))_j d_{0,j} \\ \text{s.t.} \quad & \ell_{x,j} \leq x_{0,j}^k + d_{0,j} \leq u_{x,j}, \\ & |d_{0,j}| \leq \rho. \end{aligned}$$

One two-dimensional LPCC for each pair i

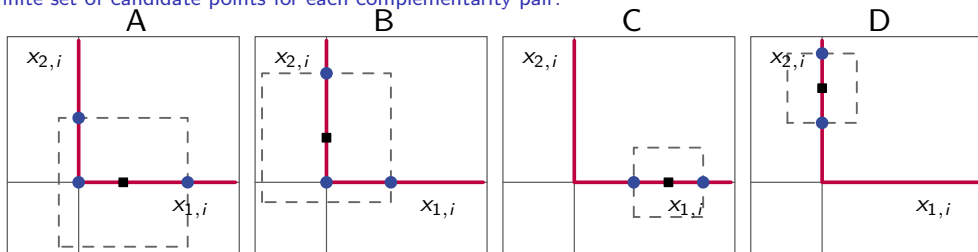
$$\begin{aligned} \min_{(d_{1,i}, d_{2,i}) \in \mathbb{R}^2} \quad & (\nabla_{x_1} f(x^k))_i d_{1,i} + (\nabla_{x_2} f(x^k))_i d_{2,i} \\ \text{s.t.} \quad & 0 \leq x_{1,i}^k + d_{1,i} \perp x_{2,i}^k + d_{2,i} \geq 0, \\ & |d_{1,i}| \leq \rho, \\ & |d_{2,i}| \leq \rho. \end{aligned}$$

The n_0 scalar LPs have closed-form solutions:

$$d_{0,j}^* = \begin{cases} \min\{u_{x,j} - x_{0,j}^k, \rho\}, & (\nabla_{x_0} f(x^k))_j < 0, \\ \max\{\ell_{x,j} - x_{0,j}^k, -\rho\}, & (\nabla_{x_0} f(x^k))_j > 0, \\ 0, & (\nabla_{x_0} f(x^k))_j = 0. \end{cases}$$

Solving each two-dimensional LPCC

Check a finite set of candidate points for each complementarity pair.



Dashed box: trust region. Square: current point. Dots: candidate points.

$$\mathcal{D}_i^A = \{(0, 0), (\rho, 0), (-x_{1,i}, 0), (-x_{1,i}, \rho)\},$$

$$x_{2,i} = 0, \quad 0 \leq x_{1,i} \leq \rho,$$

$$\mathcal{D}_i^B = \{(0, 0), (\rho, -x_{2,i}), (0, -x_{2,i}), (0, \rho)\},$$

$$x_{1,i} = 0, \quad 0 < x_{2,i} \leq \rho,$$

$$\mathcal{D}_i^C = \{(0, 0), (-\rho, 0), (\rho, 0)\},$$

$$x_{2,i} = 0, \quad x_{1,i} > \rho,$$

$$\mathcal{D}_i^D = \{(0, 0), (0, -\rho), (0, \rho)\},$$

$$x_{1,i} = 0, \quad x_{2,i} > \rho.$$

Select the candidate with the smallest linearized objective:

$$(d_{1,i}^*, d_{2,i}^*) \in \arg \min_{(a,b) \in \mathcal{D}_i} ((\nabla_{x_1} f(x))_i a + (\nabla_{x_2} f(x))_i b).$$

Outline of the lecture

1. Overview of MPCC methods

2. Regularization and penalty methods

3. Combinatorial and active-set methods

A sequential linearization method for bound-constrained MPCCs

A sequential LPCC–EQP method

MPECopt: an algorithm for computing a B-stationary points

Sequential quadratic programming with complementarity constraints

A sequential LPCC method

First method to use LPCC [Leyffer and Munson, 2007], more general than previous method [Kirches et al., 2022]

1. Compute a step d^k by solving

Full LPCC(x^k, ρ^k)

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \nabla f(x^k)^\top d \\ \text{s.t.} \quad & g(x^k) + \nabla g(x^k)^\top d = 0, \\ & h(x^k) + \nabla h(x^k)^\top d \geq 0, \\ & 0 \leq x_1^k + d_1 \perp x_2^k + d_2 \geq 0, \\ & \|d\|_\infty \leq \rho^k. \end{aligned}$$

2. Test the trial point $x^k + d^k$:

$$(x^{k+1}, \rho^{k+1}) = \begin{cases} (x^k + d^k, 2\rho^k), & \text{if the step is accepted,} \\ (x^k, \rho^k/2), & \text{otherwise.} \end{cases}$$

- ▶ The LPCC step resembles steepest descent. Can it be accelerated?
- ▶ When should we accept $x^k + d^k$? (Globalization strategy, filter trust-region for MPCCs [Leyffer and Munson, 2007])
- ▶ How do we solve the **nontrivial** LPCC?

Equality constrained quadratic program (EQP)

Given the LPCC step d^k , and guess $\bar{x} = x^k + d^k$ estimate the active sets:

$$\mathcal{A}(\bar{x}) = \{j \mid h_j(x^k) + \nabla h_j(x^k)^\top d^k = 0\},$$

$$\mathcal{I}_1(\bar{x}) = \mathcal{I}_{0+}(\bar{x}) \cup \mathcal{D}_1,$$

$$\mathcal{I}_2(\bar{x}) = \mathcal{I}_{+0}(\bar{x}) \cup \mathcal{D}_2,$$

$$\mathcal{D}_1 = \{i \in \mathcal{I}_{00}(\bar{x}) \mid \nabla_{x_{1,i}} f(\bar{x}) \geq \nabla_{x_{2,i}} f(\bar{x})\}, \mathcal{D}_2 = \mathcal{I}_{00}(\bar{x}) \setminus \mathcal{D}_1.$$

EQP(d^k)

$$\min_{s \in \mathbb{R}^n} \quad \nabla f(x^k)^\top s + \frac{1}{2} s^\top H^k s$$

$$\text{s.t.} \quad g(x^k) + \nabla g(x^k)^\top s = 0,$$

$$h_j(x^k) + \nabla h_j(x^k)^\top s = 0,$$

$$x_{1,i}^k + s_{1,i} = 0,$$

$$x_{2,i}^k + s_{2,i} = 0,$$

$$j \in \mathcal{A}(\bar{x}),$$

$$i \in \mathcal{I}_1(\bar{x}),$$

$$i \in \mathcal{I}_2(\bar{x}).$$

The solution s^k gives the second-order step.

Global convergence to B-stationarity

Assumptions

- ▶ MPCC-MFCQ holds.
- ▶ The iterates $\{x^k\}$ remain in a compact set.
- ▶ f , g , and h are twice continuously differentiable.

Theorem (Global convergence of SLPEC, [Leyffer and Munson, 2007])

One of the following occurs:

1. *The restoration phase fails to find a feasible point.*
2. *$d = 0$ solves the LPCC, so the current point is B-stationary.*
3. *The method generates an infinite sequence with a B-stationary limit.*

Outline of the lecture

1. Overview of MPCC methods

2. Regularization and penalty methods

3. Combinatorial and active-set methods

A sequential linearization method for bound-constrained MPCCs

A sequential LPCC–EQP method

MPECopt: an algorithm for computing a B-stationary points

Sequential quadratic programming with complementarity constraints

Algorithm for computing B-stationary point of MPECs

Similar to previous two algorithms, but solves BNLPs instead of BQP/EQPs

Full LPCC(x^k, ρ)

$$\min_{d \in \mathbb{R}^n} \quad \nabla f(x^k)^\top d$$

$$\text{s.t.} \quad g(x^k) + \nabla g(x^k)^\top d = 0,$$

$$h(x^k) + \nabla h(x^k)^\top d \geq 0,$$

$$0 \leq x_{1,i}^k + d_{1,i} \perp x_{2,i}^k + d_{2,i} \geq 0, \forall i,$$

$$\|d\|_\infty \leq \rho,$$

Algorithm for computing B-stationary point of MPECs

Similar to previous two algorithms, but solves BNLPs instead of BQP/EQPs

Full LPCC(x^k, ρ)

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \nabla f(x^k)^\top d \\ \text{s.t.} \quad & \mathbf{g}(x^k) + \nabla \mathbf{g}(x^k)^\top d = 0, \\ & \mathbf{h}(x^k) + \nabla \mathbf{h}(x^k)^\top d \geq 0, \\ & 0 \leq \mathbf{x}_{1,i}^k + d_{1,i} \perp \mathbf{x}_{2,i}^k + d_{2,i} \geq 0, \forall i, \\ & \|d\|_\infty \leq \rho, \end{aligned}$$

BNLP(x^k)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & \mathbf{g}(x) = 0, \\ & \mathbf{h}(x) \geq 0, \\ & x_{1,i} = 0, \ x_{2,i} \geq 0, \ i \in \mathcal{I}_1^k, \\ & x_{1,i} \geq 0, \ x_{2,i} = 0, \ i \in \mathcal{I}_2^k. \end{aligned}$$

Algorithm for computing B-stationary point of MPECs

Similar to previous two algorithms, but solves BNLPs instead of BQP/EQPs

Full LPCC(x^k, ρ)

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \nabla f(x^k)^\top d \\ \text{s.t.} \quad & \mathbf{g}(x^k) + \nabla \mathbf{g}(x^k)^\top d = 0, \\ & \mathbf{h}(x^k) + \nabla \mathbf{h}(x^k)^\top d \geq 0, \\ & 0 \leq \mathbf{x}_{1,i}^k + d_{1,i} \perp \mathbf{x}_{2,i}^k + d_{2,i} \geq 0, \forall i, \\ & \|d\|_\infty \leq \rho, \end{aligned}$$

BNLP(x^k)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & \mathbf{g}(x) = 0, \\ & \mathbf{h}(x) \geq 0, \\ & x_{1,i} = 0, \ x_{2,i} \geq 0, \ i \in \mathcal{I}_1^k, \\ & x_{1,i} \geq 0, \ x_{2,i} = 0, \ i \in \mathcal{I}_2^k. \end{aligned}$$

Main steps:

1. Solve LPCC(x^k, ρ) to determine the active set for BNLP($x^k + d$) – or verify B-stationarity. **Don't use d for iterate update.**

Algorithm for computing B-stationary point of MPECs

Similar to previous two algorithms, but solves BNLPS instead of BQP/EQPs

Full LPCC(x^k, ρ)

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \nabla f(x^k)^\top d \\ \text{s.t.} \quad & \mathbf{g}(x^k) + \nabla \mathbf{g}(x^k)^\top d = 0, \\ & \mathbf{h}(x^k) + \nabla \mathbf{h}(x^k)^\top d \geq 0, \\ & 0 \leq \mathbf{x}_{1,i}^k + d_{1,i} \perp \mathbf{x}_{2,i}^k + d_{2,i} \geq 0, \forall i, \\ & \|d\|_\infty \leq \rho, \end{aligned}$$

BNLP(x^k)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & \mathbf{g}(x) = 0, \\ & \mathbf{h}(x) \geq 0, \\ & x_{1,i} = 0, \quad x_{2,i} \geq 0, \quad i \in \mathcal{I}_1^k, \\ & x_{1,i} \geq 0, \quad x_{2,i} = 0, \quad i \in \mathcal{I}_2^k. \end{aligned}$$

Main steps:

1. Solve LPCC(x^k, ρ) to determine the active set for BNLP($x^k + d$) – or verify B-stationarity. **Don't use d for iterate update.**
2. Solve BNLP($x^k + d$) for accuracy: if decrease in objective accept step, else: resolve LPCC with smaller ρ .

Outline of the MPECopt algorithm

Algorithm 1 Algorithm Outline

Input: $x^0, \rho^0 \in [\rho^{\text{lb}}, \rho^{\text{ub}}]$, $\gamma^{\text{L}} \in (0, 1)$, $\gamma^{\text{U}} > 1$

- 1 Find a feasible point $x \in \Omega$ of the MPCC (1). (Phase I)
 - 2 **for** $k = 0, \dots$ **do** // Major loop
 - 3 Solved the branch NLP with some partition of the index set, e.g. $\mathcal{I}_1(x)$, $\mathcal{I}_2(x)$
 - 4 Solve an LPCC to verify B-stationary or to give a better $\mathcal{I}_1(x^+)$ and $\mathcal{I}_2(x^+)$
 - 5 **Break** If B-stationary (stationary point found)
-

- ▶ Algorithm iterates over active sets (feasible NLP subproblems).
- ▶ Finds descend direction if not B-stationary.
- ▶ Alternates between LPCC and BranchNLP.

Phase I – finding feasible points

Crossover strategy (from NLP relaxation to LPEC)

1. Use regularization or penalty based method with σ^k (NLP solve)
2. If $\|\text{diag}(x_1)x_2\|_\infty < \rho^0$, solve $\text{LPCC}(x^*(\tau^k), \rho^0)$.
3. Solve $\text{BranchNLP}(x^*(\sigma^k) + d)$, if successful, feasible point found.
4. If not, reduce σ^k and go to 1.

Phase I – finding feasible points

Crossover strategy (from NLP relaxation to LPEC)

1. Use regularization or penalty based method with σ^k (NLP solve)
2. If $\|\text{diag}(x_1)x_2\|_\infty < \rho^0$, solve $\text{LPCC}(x^*(\tau^k), \rho^0)$.
3. Solve $\text{BranchNLP}(x^*(\sigma^k) + d)$, if successful, feasible point found.
4. If not, reduce σ^k and go to 1.

Theorem (Feasibility (informal), [Nurkanović and Leyffer, 2026])

Under suitable technical assumptions and MPCC-MFCQ, if $x(\sigma^k)$ is close enough to a feasible point of the MPCC, then for a sufficiently large ρ , every feasible point d of $\text{LPCC}(x^(\sigma^k), \rho^0)$ predicts a feasible BranchNLP.*

Scholtes' global relaxation method

The easiest to implement and the most efficient regularization method [Scholtes, 2001].

$\text{Reg}(\sigma^k)$

$$\min_{x \in \mathbb{R}^n} f(x)$$

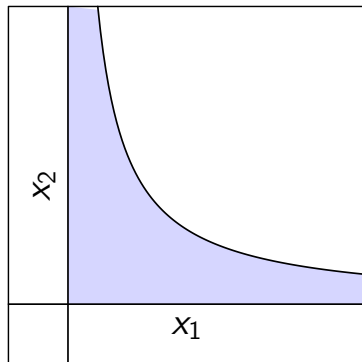
$$\text{s.t.} \quad g(x) = 0,$$

$$h(x) \geq 0,$$

$$x_1 \geq 0,$$

$$x_2 \geq 0,$$

$$x_{1,i}x_{2,i} \leq \sigma^k, \quad i = 1, \dots, m.$$



Scholtes' global relaxation method

The easiest to implement and the most efficient regularization method [Scholtes, 2001].

$\text{Reg}(\sigma^k)$

$$\min_{x \in \mathbb{R}^n} f(x)$$

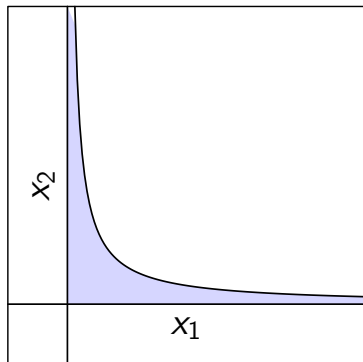
$$\text{s.t.} \quad g(x) = 0,$$

$$h(x) \geq 0,$$

$$x_1 \geq 0,$$

$$x_2 \geq 0,$$

$$x_{1,i}x_{2,i} \leq \sigma^k, \quad i = 1, \dots, m.$$



Scholtes' global relaxation method

The easiest to implement and the most efficient regularization method [Scholtes, 2001].

$\text{Reg}(\sigma^k)$

$$\min_{x \in \mathbb{R}^n} f(x)$$

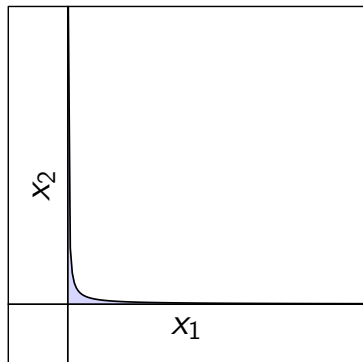
$$\text{s.t.} \quad g(x) = 0,$$

$$h(x) \geq 0,$$

$$x_1 \geq 0,$$

$$x_2 \geq 0,$$

$$x_{1,i}x_{2,i} \leq \sigma^k, \quad i = 1, \dots, m.$$



Scholtes' global relaxation method

The easiest to implement and the most efficient regularization method [Scholtes, 2001].

$\text{Reg}(\sigma^k)$

$$\min_{x \in \mathbb{R}^n} f(x)$$

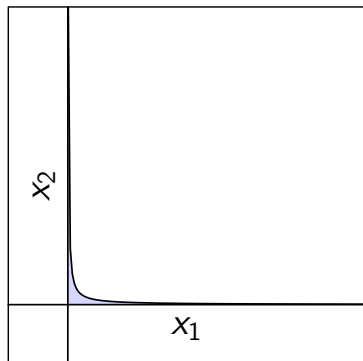
$$\text{s.t.} \quad g(x) = 0,$$

$$h(x) \geq 0,$$

$$x_1 \geq 0,$$

$$x_2 \geq 0,$$

$$x_{1,i}x_{2,i} \leq \sigma^k, \quad i = 1, \dots, m.$$



Eventually BranchNLP must be feasible for feasible MPCCs ... or detect infeasibility!

MPECopt: An algorithm for computing B-stationary points

Finite termination of all loops proven in [Nurkanović and Leyffer, 2026]

Algorithm 2 MPECopt

Input: $x^0, \rho^0 \in [\rho^{\text{lb}}, \rho^{\text{ub}}]$, $\gamma^{\text{L}} \in (0, 1)$, $\gamma^{\text{U}} > 1$

```
6 Call Phase I Algorithm to find a feasible point  $x^0 \in \Omega$ 
   for  $k = 0, \dots$  do                                     // Major loop
7     Possibly reset trust region radius  $\rho^{k,0} \in [\rho^{\text{lb}}, \rho^{\text{ub}}]$ 
8     for  $l = 0, \dots$  do                                     // Minor loop
9         Solve LPEC( $x^k, \rho^{k,l}$ ) with TR-radius  $\rho^{k,l}$  for  $d^{k,l}$ 
10        if  $d^{k,l} = 0$  then
11            terminate                                     // (B-stationary point found)
12        else if Active set changed then
13            Solve BranchNLP( $x^k + d^{k,l}$ ) for  $x^{k,l}$ 
14            if  $f(x^{k,l}) < f(x^k)$  then
15                 $x^{k+1} = x^{k,l}$ ;  $\rho^{k,l+1} = \gamma^{\text{U}} \rho^{k,l}$ ;           // step accepted
16            else  $\rho^{k,l+1} = \gamma^{\text{L}} \rho^{k,l}$                        // reduce TR-radius
17        else  $\rho^{k,l+1} = \gamma^{\text{L}} \rho^{k,l}$                        // reduce TR-radius
```

Algorithm for computing B-stationary point of an MPCC

Assumptions

- ▶ MPCC-MFCQ holds.
- ▶ The feasible set is compact.
- ▶ The functions f , g , and h are twice continuously differentiable.
- ▶ An NLP solver with a restoration phase is available.

Theorem (Global convergence of MPECopt [Nurkanović and Leyffer, 2026])

One of the following occurs:

1. *The restoration phase fails to find a feasible point.*
2. *A B-stationary point is found after finitely many NLP and LPCC solves.*
3. *The method generates an infinite sequence with a B-stationary limit point^{*}.*

^{*}The third outcome cannot occur if all stationary points are isolated.

LPCC(x^k, ρ) as MILP

$$\min_{\substack{d \in \mathbb{R}^n, \\ y \in \{0,1\}^m}} \nabla f(x^k)^\top d \quad (3a)$$

$$\text{s.t.} \quad g(x^k) + \nabla g(x^k)^\top d = 0, \quad (3b)$$

$$h(x^k) + \nabla h(x^k)^\top d \geq 0, \quad (3c)$$

$$0 \leq x_1^k + d_1 \leq yM, \quad (3d)$$

$$0 \leq x_2^k + d_2 \leq (e - y)M, \quad (3e)$$

$$\|d\|_\infty \leq \rho. \quad (3f)$$

- Trust region $\|d\| \leq \rho$ makes feasible set compact.
- At feasible points, ρ can be arbitrarily small – very tight relaxations.
- BranchNLP index sets:

$$\mathcal{I}_1(x^k + d) = \{i \mid y_i = 0\},$$

$$\mathcal{I}_2(x^k + d) = \{i \mid y_i = 1\}.$$

LPCC(x^k, ρ) as MILP

$$\min_{\substack{d \in \mathbb{R}^n, \\ y \in \{0,1\}^m}} \nabla f(x^k)^\top d \quad (3a)$$

$$\text{s.t.} \quad g(x^k) + \nabla g(x^k)^\top d = 0, \quad (3b)$$

$$h(x^k) + \nabla h(x^k)^\top d \geq 0, \quad (3c)$$

$$0 \leq x_1^k + d_1 \leq yM, \quad (3d)$$

$$0 \leq x_2^k + d_2 \leq (e - y)M, \quad (3e)$$

$$\|d\|_\infty \leq \rho. \quad (3f)$$

- Trust region $\|d\| \leq \rho$ makes feasible set compact.
- At feasible points, ρ can be arbitrarily small – very tight relaxations.
- BranchNLP index sets:

$$\mathcal{I}_1(x^k + d) = \{i \mid y_i = 0\},$$

$$\mathcal{I}_2(x^k + d) = \{i \mid y_i = 1\}.$$

Proposition ([Nurkanović and Leyffer, 2026])

Let $x^* \in \Omega$ be a S -stationary point of the MPCC (1). For $\rho < \frac{M}{2}$, $d = 0$ is a global minimizer of both the MILP and its relaxation with $y_i \in [0, 1]$.

Often no need for optimality in LPCCs

- ▶ Finding a feasible BranchNLP via an LPCC in Phase I – find LPCC feasible point.
- ▶ Finding a descent direction in Phase II – find feasible LPCC point with $\nabla f(x^k)^\top d < 0$.
- ▶ If point S-stationary, solving the reduced LPCC-MILP reduces to solving the root node LP.
- ▶ Only need optimality for verifying B-stationarity, – with small ρ : tight MILP relaxation.
- ▶ Special case: only bound and complementarity constraints – linear complexity [Kirches et al., 2022].

The MPCC is also just a MINLP

MPCC

$$\min_{x \in \mathbb{R}^n} f(x) \quad (4a)$$

$$\text{s.t.} \quad g(x) = 0, \quad (4b)$$

$$h(x) \geq 0, \quad (4c)$$

$$0 \leq x_1 \perp x_2 \geq 0. \quad (4d)$$

Equivalent big- M MINLP

$$\min_{\substack{x \in \mathbb{R}^n, \\ y \in \{0,1\}^m}} f(x) \quad (5a)$$

$$\text{s.t.} \quad g(x) = 0, \quad (5b)$$

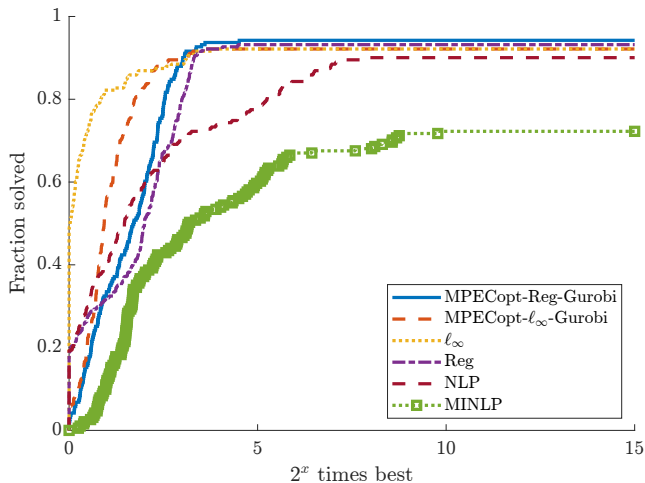
$$h(x) \geq 0, \quad (5c)$$

$$0 \leq x_1 \leq My, \quad (5d)$$

$$0 \leq x_2 \leq M(e - y). \quad (5e)$$

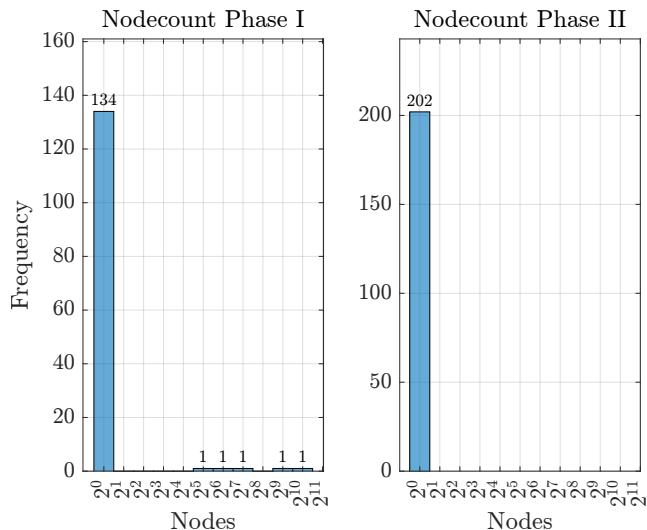
- ▶ No special considerations as for LPCCs, no obvious early terminations.
- ▶ $M > 0$ is a valid upper bound, can use individual bounds $M_{1,i}$ and $M_{2,i}$ for each $x_{1,i}$ and $x_{2,i}$.
- ▶ The resulting nonconvex MINLP can be solved with any general-purpose nonconvex MINLP method, see Lecture 9.

MPECopt on macMPEC



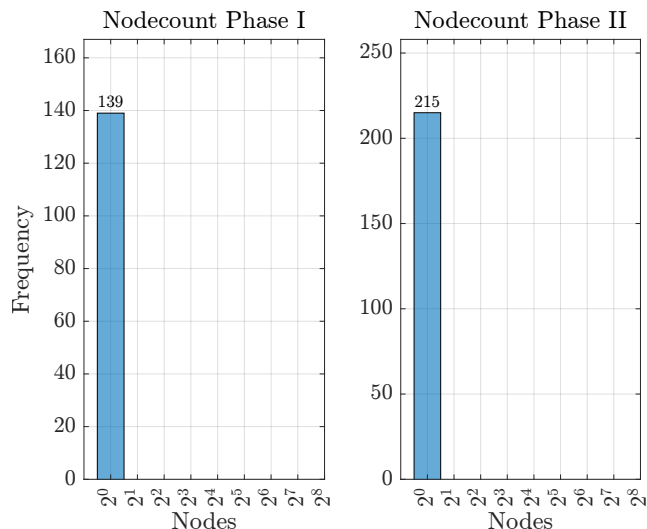
Experiments via  github.com/nosnoc/mpecopt:
Ipopt [Wächter and Biegler, 2006]+Gurobi via CasADi [Andersson et al., 2019].

Effort in solving LPCCs - solving LPCC to global optimality



Numbers of LPs solved in LPCC-MILPs on the MacMPEC test set.

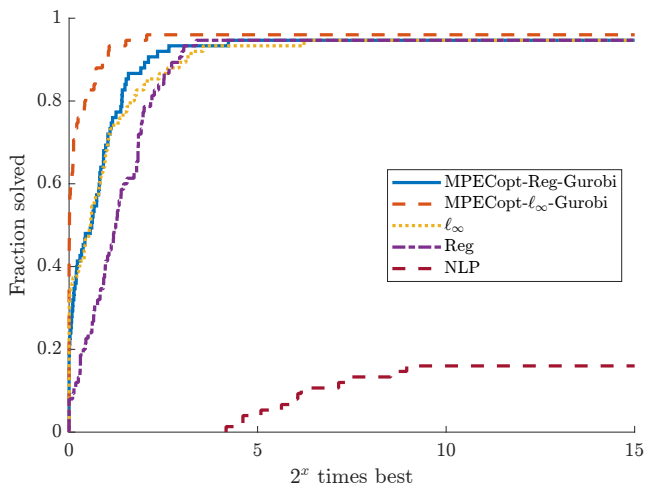
Effort in solving LPCCs - early termination of LPECs



Numbers of LPs solved in LPCC-MILPs on the MacMPEC test set.

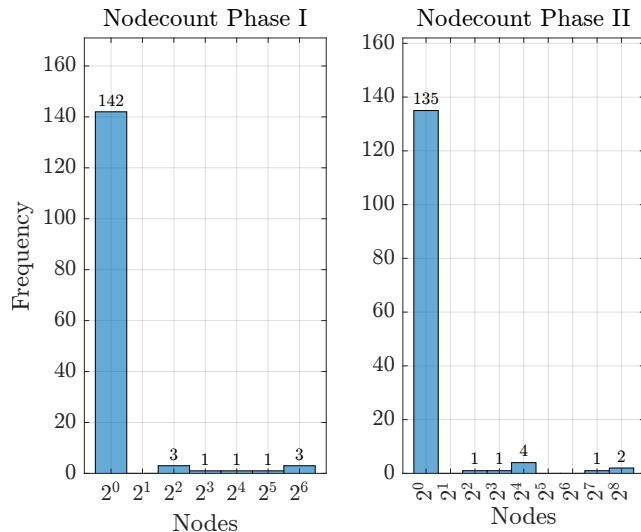
Randomly generated degenerate nonlinear MPECs

All problems violate MPEC-LICQ



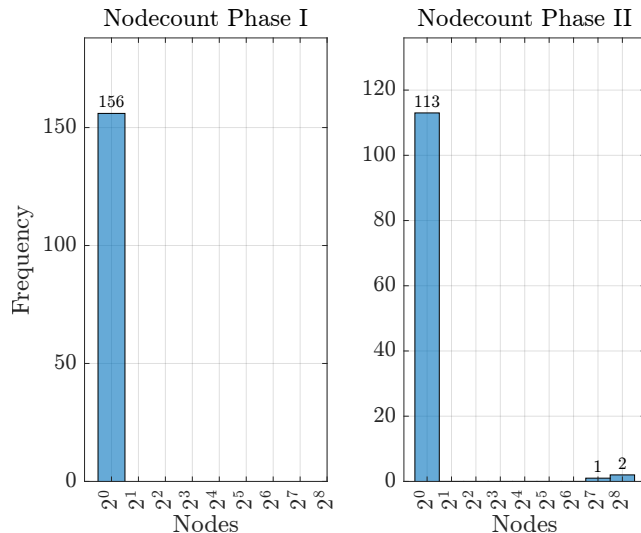
Experiments via  github.com/nosnoc/mpecopt:
Ipopt [Wächter and Biegler, 2006]+Gurobi via CasADi [Andersson et al., 2019].

Effort in solving LPCCs – solving LPCC to global optimality



Numbers of LPs solved in LPCC-MILPs on the random nonlinear test set.

Effort in solving LPCCs - early termination of LPCCs



Numbers of LPs solved in LPCC-MILPs on the random nonlinear test set.

Outline of the lecture

1. Overview of MPCC methods

2. Regularization and penalty methods

3. Combinatorial and active-set methods

A sequential linearization method for bound-constrained MPCCs

A sequential LPCC–EQP method

MPECopt: an algorithm for computing a B-stationary points

Sequential quadratic programming with complementarity constraints

Refresher: the SQP method

For the lecture's NLP convention $g(x) = 0$, $h(x) \geq 0$, use

$$\mathcal{L}(x, \lambda, \mu) = f(x) - \lambda^\top g(x) - \mu^\top h(x), \quad H^k \approx \nabla_{xx}^2 \mathcal{L}(x^k, \lambda^k, \mu^k).$$

Structural linearization of its KKT conditions, with $\Delta x = x^{k+1} - x^k$:

$$\nabla f(x^k) - \nabla g(x^k) \lambda^{k+1} - \nabla h(x^k) \mu^{k+1} + H^k \Delta x = 0, \quad (6a)$$

$$g(x^k) + \nabla g(x^k)^\top \Delta x = 0, \quad (6b)$$

$$0 \leq \mu^{k+1} \perp h(x^k) + \nabla h(x^k)^\top \Delta x \geq 0. \quad (6c)$$

Refresher: the SQP method

For the lecture's NLP convention $g(x) = 0$, $h(x) \geq 0$, use

$$\mathcal{L}(x, \lambda, \mu) = f(x) - \lambda^\top g(x) - \mu^\top h(x), \quad H^k \approx \nabla_{xx}^2 \mathcal{L}(x^k, \lambda^k, \mu^k).$$

Structural linearization of its KKT conditions, with $\Delta x = x^{k+1} - x^k$:

$$\nabla f(x^k) - \nabla g(x^k) \lambda^{k+1} - \nabla h(x^k) \mu^{k+1} + H^k \Delta x = 0, \quad (6a)$$

$$g(x^k) + \nabla g(x^k)^\top \Delta x = 0, \quad (6b)$$

$$0 \leq \mu^{k+1} \perp h(x^k) + \nabla h(x^k)^\top \Delta x \geq 0. \quad (6c)$$

These are the KKT conditions of the SQP subproblem:

SQP subproblem

$$\begin{aligned} \min_{\Delta x \in \mathbb{R}^n} \quad & \nabla f(x^k)^\top \Delta x + \frac{1}{2} \Delta x^\top H^k \Delta x \\ \text{s.t.} \quad & g(x^k) + \nabla g(x^k)^\top \Delta x = 0, \\ & h(x^k) + \nabla h(x^k)^\top \Delta x \geq 0. \end{aligned}$$

A zero step satisfying these conditions gives an NLP KKT point at x^k .

MPCC first-order necessary optimality conditions

Alternative definition of B-stationarity

At a feasible $\bar{x}$, partition $\mathcal{I}_{00}(\bar{x})$ into $\mathcal{D}_1 \cup \mathcal{D}_2$ and set

$$\mathcal{I}_1 = \mathcal{I}_{0+}(\bar{x}) \cup \mathcal{D}_1, \quad \mathcal{I}_2 = \mathcal{I}_{+0}(\bar{x}) \cup \mathcal{D}_2.$$

Under MPCC-LICQ, $\bar{x}$ is B-stationary if and only if it satisfies the KKT conditions of every corresponding BNLPP. With

$$\mathcal{L}^{\text{MPCC}}(x, \lambda, \mu, \nu, \xi) = f(x) - \lambda^\top g(x) - \mu^\top h(x) - \nu^\top x_1 - \xi^\top x_2,$$

each branch requires multipliers such that

$$\begin{aligned} \nabla_x \mathcal{L}^{\text{MPCC}}(\bar{x}, \lambda, \mu, \nu, \xi) &= 0, \\ g(\bar{x}) &= 0, \\ 0 &\leq \mu \perp h(\bar{x}) \geq 0, \\ \bar{x}_{1,i} &= 0, \quad 0 \leq \xi_i \perp \bar{x}_{2,i} \geq 0 && (i \in \mathcal{I}_1), \\ \bar{x}_{2,i} &= 0, \quad 0 \leq \nu_i \perp \bar{x}_{1,i} \geq 0 && (i \in \mathcal{I}_2). \end{aligned}$$

Here ν_i is free on $\mathcal{I}_1$ and ξ_i is free on $\mathcal{I}_2$. Under MPCC-LICQ the branch multipliers coincide and are unique.

The SQPCC method

Sequential quadratic programming with complementarity constraints [Scholtes, 2004], extended convergence theory [Nurkanović, 2026]

Model all branch QPs together by retaining complementarity in the SQP subproblem. Let

$$H^k \approx \nabla_{xx}^2 \mathcal{L}^{\text{MPCC}}(x^k, \lambda^k, \mu^k, \nu^k, \xi^k).$$

Quadratic program with complementarity constraints (QPCC)

$$\begin{aligned} \min_{\Delta x \in \mathbb{R}^n} \quad & \nabla f(x^k)^\top \Delta x + \frac{1}{2} \Delta x^\top H^k \Delta x \\ \text{s.t.} \quad & g(x^k) + \nabla g(x^k)^\top \Delta x = 0, \\ & h(x^k) + \nabla h(x^k)^\top \Delta x \geq 0, \\ & 0 \leq x_1^k + \Delta x_1 \perp x_2^k + \Delta x_2 \geq 0. \end{aligned}$$

The next primal iterate is $x^{k+1} = x^k + \Delta x^k$, the QPCC supplies updated multiplier estimates.

Outline of the lecture

1. Overview of MPCC methods

2. Regularization and penalty methods

Regularization methods

Penalty methods

3. Combinatorial and active-set methods

A sequential linearization method for bound-constrained MPCCs

A sequential LPCC–EQP method

MPECopt: an algorithm for computing a B-stationary points

Sequential quadratic programming with complementarity constraints

Summary

- ▶ Regularization and penalty methods solve a sequence of standard NLP subproblems and are relatively easy to implement.
- ▶ Tailored implementations such as CCOpt lead a great speed up in regularization methods.
- ▶ Their limit points need not be B-stationary, although this issue appears uncommon in practice.
- ▶ A globally solved LPCC can certify B-stationarity when its zero step is optimal, even if MPCC-LICQ fails.
- ▶ Active-set methods combine LPCC steps with branch-NLP solves. LPCCs do not need to be solved to optimality unless we are the solution. Under suitable assumptions, they converge to B-stationary points.
- ▶ LPCCs can be formulated as MILPs; a small trust region often tightens the formulation and reduces branching.

Cited references I

-  Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., and Diehl, M. (2019). CasADi – a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36.
-  Anitescu, M. (2005a). Global convergence of an elastic mode approach for a class of mathematical programs with complementarity constraints. *SIAM Journal on Optimization*, 16:120–145.
-  Anitescu, M. (2005b). On using the elastic mode in nonlinear programming approaches to mathematical programs with complementarity constraints. *SIAM Journal on Optimization*, 15(4):1203–1236.
-  DeMiguel, V., Friedlander, M. P., Nogales, F. J., and Scholtes, S. (2005). A two-sided relaxation scheme for mathematical programs with equilibrium constraints. *SIAM Journal on Optimization*, 16(2):587–609.

Cited references II

-  Flegel, M. L. and Kanzow, C. (2005).
Abadie-type constraint qualification for mathematical programs with equilibrium constraints.
Journal of Optimization Theory and Applications, 124(3):595–614.
-  Fletcher, R. and Leyffer, S. (2002).
Nonlinear programming without a penalty function.
Mathematical Programming, 91:239–269.
-  Fukushima, M. and Tseng, P. (2002).
An implementable active-set algorithm for computing a b-stationary point of a mathematical program with linear complementarity constraints.
SIAM Journal on Optimization, 12(3):724–739.
-  Giallombardo, G. and Ralph, D. (2008).
Multiplier convergence in trust-region methods with application to convergence of decomposition methods for mpecs.
Mathematical Programming, 112(2):335–369.

Cited references III



Guo, L. and Deng, Z. (2022).

A new augmented lagrangian method for mpccs—theoretical and numerical comparison with existing augmented lagrangian methods.

Mathematics of Operations Research, 47(2):1229–1246.



Hoheisel, T., Kanzow, C., and Schwartz, A. (2013).

Theoretical and numerical comparison of relaxation methods for mathematical programs with complementarity constraints.

Mathematical Programming, 137(1):257–288.



Izmailov, A. F. and Solodov, M. V. (2008).

An active-set newton method for mathematical programs with complementarity constraints.

SIAM Journal on Optimization, 19(3):1003–1027.



Jiang, H. and Ralph, D. (1999).

Qpecgen, a matlab generator for mathematical programs with quadratic objectives and affine variational inequality constraints.

Computational Optimization and Applications, 13:25–59.

Cited references IV

-  Kadrani, A., Dussault, J.-P., and Benchakroun, A. (2009).
A new regularization scheme for mathematical programs with complementarity constraints.
SIAM Journal on Optimization, 20(1):78–103.
-  Kanzow, C. and Schwartz, A. (2013).
A new regularization method for mathematical programs with complementarity constraints with strong convergence properties.
SIAM Journal on Optimization, 23(2):770–798.
-  Kanzow, C. and Schwartz, A. (2015).
The price of inexactness: convergence properties of relaxation methods for mathematical programs with complementarity constraints revisited.
Mathematics of Operations Research, 40(2):253–275.
-  Kazi, S. R., Thombre, M., and Biegler, L. (2024).
Globally convergent method for optimal control of hybrid dynamical systems.
IFAC-PapersOnLine, 58(14):868–873.

Cited references V

-  Kirches, C., Larson, J., Leyffer, S., and Manns, P. (2022).
Sequential linearization method for bound-constrained mathematical programs with complementarity constraints.
SIAM Journal on Optimization, 32(1):75–99.
-  Leyffer, S. (2000).
Macmpec: Ampl collection of mpecs," .
Argonne National Laboratory.
-  Leyffer, S., López-Calva, G., and Nocedal, J. (2006).
Interior methods for mathematical programs with complementarity constraints.
SIAM Journal on Optimization, 17(1):52–77.
-  Leyffer, S. and Munson, T. S. (2007).
A globally convergent filter method for mpecs.
Preprint ANL/MCS-P1457-0907, Argonne National Laboratory, Mathematics and Computer Science Division.

Cited references VI

-  Lin, G.-H. and Fukushima, M. (2005).
A modified relaxation scheme for mathematical programs with complementarity constraints.
Annals of Operations Research, 133(1):63–84.
-  Lin, G.-H. and Fukushima, M. (2006).
Hybrid approach with active set identification for mathematical programs with complementarity constraints.
Journal of optimization theory and applications, 128(1):1–28.
-  Liu, X., Perakis, G., and Sun, J. (2006).
A robust SQP method for mathematical programs with linear complementarity constraints.
Computational Optimization and Applications, 34(1):5–33.
-  Luo, Z., Pang, J., and Ralph, D. (1996).
Mathematical Programs with Equilibrium Constraints.
Cambridge University Press, Cambridge.

Cited references VII

 Nurkanović, A. (2026).
Local convergence results for sequential quadratic programming with complementarity constraints.
arXiv preprint arXiv:2604.18192.

 Nurkanović, A. and Leyffer, S. (2026).
A globally convergent method for computing b-stationary points of mathematical programs with equilibrium constraints.
Computational Optimization and Applications (accepted for publication).

 Nurkanović, A., Pozharskiy, A., and Diehl, M. (2024).
Solving mathematical programs with complementarity constraints arising in nonsmooth optimal control.
Vietnam Journal of Mathematics, pages 1–39.

 Pozharskiy, A., Pacaud, F., Diehl, M., and Nurkanović, A. (2026).
CCOpt: an open-source solver for large-scale mathematical programs with complementarity constraints.
arXiv preprint.

Cited references VIII

-  Raghunathan, A. U. and Biegler, L. T. (2005).
An interior point method for mathematical programs with complementarity constraints (mpccs).
SIAM Journal on Optimization, 15(3):720–750.
-  Ralph, D. and Wright, S. J. (2004).
Some properties of regularization and penalization schemes for mpecs.
Optimization Methods and Software, 19(5):527–556.
-  Scholtes, S. (2001).
Convergence properties of a regularization scheme for mathematical programs with complementarity constraints.
SIAM Journal on Optimization, 11(4):918–936.
-  Scholtes, S. (2004).
Nonconvex structures in nonlinear programming.
Operations Research, 52(3):368–383.
-  Scholtes, S. and Stöhr, M. (1999).
Exact penalization of mathematical programs with equilibrium constraints.
SIAM Journal on Control and Optimization, 37(2):617–652.

Cited references IX

-  Steffensen, S. and Ulbrich, M. (2010).
A new relaxation scheme for mathematical programs with equilibrium constraints.
SIAM Journal on Optimization, 20(5):2504–2539.
-  Wächter, A. and Biegler, L. T. (2006).
On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming.
Mathematical Programming, 106(1):25–57.