Revisiting Design Aspects of a QP Solver for WORHP

7th International Conference on Astrodynamics Tools and Techniques (ICATT)

Marcel Jacobse, Christof Büskens

University of Bremen, Center for Industrial Mathematics

November 9, 2018

Contents

1 Sequential Quadratic Programming Approach of WORHP

2 Solving WORHP's Quadratic Subproblems

3 Efficient Formulation of WORHP's Subproblems

4 Numerical Results with the new QP Solver

WORHP

We Optimize Really Huge Problems

- Solver for large-scale nonlinear optimization
- Interfaces for C, C++, Fortran, AMPL, ...
- Free for academics, www.worhp.de
- Method[1]: Sequential Quadratic Programming (SQP)

[1] Recently: New penalty interior point method, developed by Renke Kuhlmann

Sequential Quadratic Programming

NLP:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & \underline{g} \leq g(x) \leq \overline{g} \end{aligned}$$

Sequential Quadratic Programming

NLP:

$$\min_{x \in \mathbb{R}^n} \quad f(x)$$
$$\text{s.t.} \quad \underline{g} \leq g(x) \leq \overline{g}$$

QP:

$$\min_{d \in \mathbb{R}^n} \quad \frac{1}{2} d^\mathsf{T} H^{[k]} d + \nabla_x f(x^{[k]})^\mathsf{T} d$$
$$\text{s.t.} \quad \underline{g} \leq g(x^{[k]}) + \nabla_x g(x^{[k]})^\mathsf{T} d \leq \overline{g}$$

- Approximate NLP with a QP in each iteration

Sequential Quadratic Programming

NLP:

$$\min_{x \in \mathbb{R}^n} \quad f(x)$$
$$\text{s.t.} \quad \underline{g} \leq g(x) \leq \overline{g}$$

QP:

$$\min_{d \in \mathbb{R}^n} \quad \frac{1}{2} d^T H^{[k]} d + \nabla_x f(x^{[k]})^T d$$
$$\text{s.t.} \quad \underline{g} \leq g(x^{[k]}) + \nabla_x g(x^{[k]})^T d \leq \overline{g}$$

- Approximate NLP with a QP in each iteration
- Use direction d to get new iterate

$$x^{[k+1]} := x^{[k]} + \alpha d$$

Sequential Quadratic Programming

NLP:

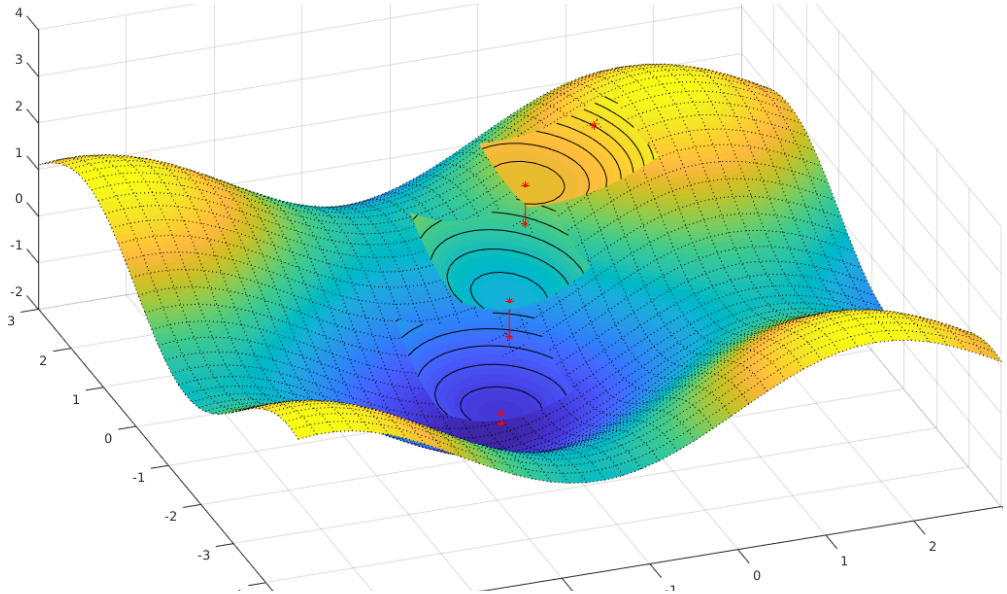
$$\min_{x \in \mathbb{R}^n} \quad f(x)$$
$$\text{s.t.} \quad \underline{g} \leq g(x) \leq \overline{g}$$

QP:

$$\min_{d \in \mathbb{R}^n} \quad \frac{1}{2} d^\mathsf{T} H^{[k]} d + \nabla_x f(x^{[k]})^\mathsf{T} d$$
$$\text{s.t.} \quad \underline{g} \leq g(x^{[k]}) + \nabla_x g(x^{[k]})^\mathsf{T} d \leq \overline{g}$$

- Approximate NLP with a QP in each iteration
- Use direction d to get new iterate

$$x^{[k+1]} := x^{[k]} + \alpha d$$



Sequential Quadratic Programming

Line Search SQP Methods

- Need to solve a QP in every iteration

Inequality constrained QPs: Solution not trivial

Common Solution Methods

- Active-Set Methods
- Interior Point Methods

Solving the Quadratic Program

QPSOL

- WORHP uses Fortran code QPSOL to solve the subproblems
- Mehrotra's predictor-corrector variant of a primal-dual interior point method

Solving the Quadratic Program

QPSOL

- WORHP uses Fortran code QPSOL to solve the subproblems
- Mehrotra's predictor-corrector variant of a primal-dual interior point method

- Experience from using WORHP in many areas
- Development of new features
- Code refactoring

Solving the Quadratic Program

QPSOL

- WORHP uses Fortran code QPSOL to solve the subproblems
- Mehrotra's predictor-corrector variant of a primal-dual interior point method

- Experience from using WORHP in many areas
- Development of new features
- Code refactoring

▶ Interest in reworked QP solver ▶ "WORHP QP" (C++)

Interior Point Method

QP in Standard Form

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \frac{1}{2} x^T Q x + c^T x \\ \text{s.t.} \quad & Ax = b, \quad x \geq 0 \end{aligned}$$

Optimality Conditions

$$\begin{aligned} Qx + c + A^T y - z &= 0 \\ Ax - b &= 0 \\ ZXe &= 0 \end{aligned}$$

Interior Point Method

QP in Standard Form

$$\min_{x \in \mathbb{R}^n} \quad \frac{1}{2} x^T Q x + c^T x - \mu \sum_{i=1}^{n} \log(x_i)$$
$$\text{s.t.} \quad Ax = b, \quad x \geq 0$$

Optimality Conditions

$$Qx + c + A^T y - z = 0$$
$$Ax - b = 0$$
$$ZXe - \mu e = 0$$

- Perturb problem with barrier parameter μ

Interior Point Method

QP in Standard Form

$$\min_{x \in \mathbb{R}^n} \quad \frac{1}{2} x^T Q x + c^T x - \mu \sum_{i=1}^{n} \log(x_i)$$
$$\text{s.t.} \quad Ax = b, \quad x \geq 0$$

Optimality Conditions

$$Qx + c + A^T y - z = 0$$
$$Ax - b = 0$$
$$ZXe - \mu e = 0$$

- Perturb problem with barrier parameter μ
- Apply Newton's method

$$\begin{pmatrix} Q & A^T & 0 \\ A & 0 & 0 \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = - \begin{pmatrix} Qx + c + A^T y - z \\ Ax - b \\ ZXe - \mu e \end{pmatrix}$$

Interior Point Method

- Apply Newton's method

$$\begin{pmatrix} Q & A^T & 0 \\ A & 0 & 0 \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = - \begin{pmatrix} Qx + c + A^T y - z \\ Ax - b \\ ZXe - \mu e \end{pmatrix}$$

Interior Point Method

- Apply Newton's method

$$\begin{pmatrix} Q & A^T & 0 \\ A & 0 & 0 \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = - \begin{pmatrix} Qx + c + A^T y - z \\ Ax - b \\ ZXe - \mu e \end{pmatrix}$$

- Take maximum stepsize $\alpha \in (0, 1]$ such that new iterates

$$x^+ := x + \alpha \Delta x > 0 \text{ and } z^+ := z + \alpha \Delta z > 0$$

Interior Point Method

- Apply Newton's method

$$\begin{pmatrix} Q & A^T & 0 \\ A & 0 & 0 \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = - \begin{pmatrix} Qx + c + A^T y - z \\ Ax - b \\ ZXe - \mu e \end{pmatrix}$$

- Take maximum stepsize $\alpha \in (0, 1]$ such that new iterates

$$x^+ := x + \alpha \Delta x > 0 \text{ and } z^+ := z + \alpha \Delta z > 0$$

- Keep iterates somewhat close to center, $x_i z_i \approx \mu$
- Let $\mu \to 0$ progressively throughout iterations

Interior Point Method

- Apply Newton's method

$$\begin{pmatrix} Q & A^T & 0 \\ A & 0 & 0 \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = - \begin{pmatrix} Qx + c + A^T y - z \\ Ax - b \\ ZXe - \mu e \end{pmatrix}$$

- Take maximum stepsize $\alpha \in (0, 1]$ such that new iterates

$$x^+ := x + \alpha \Delta x > 0 \text{ and } z^+ := z + \alpha \Delta z > 0$$

- Keep iterates somewhat close to center, $x_i z_i \approx \mu$
- Let $\mu \to 0$ progressively throughout iterations
  - Based on $\mu_{\text{curr}} = \frac{x^T z}{n}$ choose $\mu = \sigma \mu_{\text{curr}}$ with $\sigma \in (0, 1)$
  - Mehrotra's method provides efficient heuristic choice for σ

Mehrotra Predictor-Corrector

1. Solve Newton-System for $\mu = 0$ ▶ predictor step Δ_{pred}

Mehrotra Predictor-Corrector

1. Solve Newton-System for $\mu = 0$ ▶ predictor step Δ_{pred}
2. Calculate average complementarity product of predicted iterates

$$\mu_{\text{pred}} := \frac{(x + \alpha_{\text{pred}} \Delta x_{\text{pred}})^T (z + \alpha_{\text{pred}} \Delta z_{\text{pred}})}{n}$$

3. Set reduction factor $\sigma := \left(\frac{\mu_{\text{pred}}}{\mu_{\text{corr}}} \right)^2$

Mehrotra Predictor-Corrector

1. Solve Newton-System for $\mu = 0$ ▶ predictor step Δ_{pred}
2. Calculate average complementarity product of predicted iterates

$$\mu_{\text{pred}} := \frac{(x + \alpha_{\text{pred}} \Delta x_{\text{pred}})^T (z + \alpha_{\text{pred}} \Delta z_{\text{pred}})}{n}$$

3. Set reduction factor $\sigma := \left(\frac{\mu_{\text{pred}}}{\mu_{\text{corr}}} \right)^2$
4. Solve Newton-System for $\mu = \sigma \mu_{\text{curr}}$ ▶ corrector step Δ_{corr}
5. Combine for final direction

$$\Delta := \Delta_{\text{pred}} + \Delta_{\text{corr}}$$

Added Extensions for new solver WORHP QP

Gondzio's Multiple Centrality Correctors

1. For an aspired stepsize $\tilde{\alpha} > \alpha_{\text{pred}}$ calculate intermediary iterates

$$\tilde{x} := x + \tilde{\alpha}\Delta x_{\text{pred}} \text{ and } \tilde{z} := z + \tilde{\alpha}\Delta z_{\text{pred}}$$

Added Extensions for new solver WORHP QP

Gondzio's Multiple Centrality Correctors

1. For an aspired stepsize $\tilde{\alpha} > \alpha_{\text{pred}}$ calculate intermediary iterates

$$\tilde{x} := x + \tilde{\alpha}\Delta x_{\text{pred}} \text{ and } \tilde{z} := z + \tilde{\alpha}\Delta z_{\text{pred}}$$

2. Calculate target $t_i := \begin{cases} \gamma\mu & \text{if } \tilde{x}_i\tilde{z}_i \leq \gamma\mu \\ \frac{1}{\gamma}\mu & \text{if } \tilde{x}_i\tilde{z}_i \geq \frac{1}{\gamma}\mu \\ \tilde{x}_i\tilde{z}_i & \text{otherwise} \end{cases}$

Added Extensions for new solver WORHP QP

Gondzio's Multiple Centrality Correctors

1. For an aspired stepsize $\tilde{\alpha} > \alpha_{\text{pred}}$ calculate intermediary iterates

$$\tilde{x} := x + \tilde{\alpha}\Delta x_{\text{pred}} \text{ and } \tilde{z} := z + \tilde{\alpha}\Delta z_{\text{pred}}$$

2. Calculate target $t_i := \begin{cases} \gamma\mu & \text{if } \tilde{x}_i\tilde{z}_i \leq \gamma\mu \\ \frac{1}{\gamma}\mu & \text{if } \tilde{x}_i\tilde{z}_i \geq \frac{1}{\gamma}\mu \\ \tilde{x}_i\tilde{z}_i & \text{otherwise} \end{cases}$

3. Use right hand side $-\left(0, 0, \tilde{X}\tilde{Z}e - t\right)$ to compute corrector direction Δ_{corr}

Added Extensions for new solver WORHP QP

Weighted Correctors

- Instead of combining $\Delta := \Delta_{\text{pred}} + \Delta_{\text{corr}}$ combine

$$\Delta := \Delta_{\text{pred}} + \omega \Delta_{\text{corr}}$$

  with a weight $\omega \in (0, 1)$ such that allowed stepsize is maximized.
- Use simple line search to avoid bi-level optimization problem

Quadratic Problem Formulation

WORHP's Subproblem

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \frac{1}{2} d^T H^{[k]} d + \nabla_x f(x^{[k]})^T d \\ \text{s.t.} \quad & \underline{x} \leq x^{[k]} + d \leq \overline{x} \\ & \underline{g} \leq g(x^{[k]}) + \nabla_x g(x^{[k]})^T d \leq \overline{g} \end{aligned}$$

Quadratic Problem Formulation

WORHP's Subproblem

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \frac{1}{2} d^T H^{[k]} d + \nabla_x f(x^{[k]})^T d \\ \text{s.t.} \quad & \underline{x} \leq x^{[k]} + d \leq \overline{x} \\ & \underline{g} \leq g(x^{[k]}) + \nabla_x g(x^{[k]})^T d \leq \overline{g} \end{aligned}$$

QPSOL's Formulation

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \frac{1}{2} x^T Q x + c^T x \\ \text{s.t.} \quad & \underline{x} \leq x \leq \overline{x} \\ & Ax = b \\ & Cx \leq d \end{aligned}$$

Quadratic Problem Formulation

WORHP's Subproblem

$$\begin{aligned} \min_{d\in\mathbb{R}^n} \quad & \frac{1}{2} d^T H^{[k]} d + \nabla_x f(x^{[k]})^T d \\ \text{s.t.} \quad & \underline{x} \le x^{[k]} + d \le \overline{x} \\ & \underline{g} \le g(x^{[k]}) + \nabla_x g(x^{[k]})^T d \le \overline{g} \end{aligned}$$

QPSOL's Formulation

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \frac{1}{2} x^T Q x + c^T x \\ \text{s.t.} \quad & \underline{x} \le x \le \overline{x} \\ & Ax = b \\ & Cx \le d \end{aligned}$$

- Formulation limitations require WORHP to:
  - Split $\nabla_x g(x^{[k]})^T$ into A and C
  - Need to duplicate constraints with $\underline{g}_i > -\infty$ and $\overline{g}_i < \infty$
  - Create extra row in A for all fixed variables $\underline{x}_i = \overline{x}_i$

Quadratic Problem Formulation

Reworked Solver WORHP QP

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \frac{1}{2} x^\mathsf{T} Q x + c^\mathsf{T} x \\ \text{s.t.} \quad & \underline{x} \leq x \leq \overline{x} \\ & \underline{b} \leq Ax \leq \overline{b} \end{aligned}$$

- No reordering into equality and inequality constraints needed
- Fixed variables are eliminated from the problem
- Constraints with both sided bounds are handled internally

Handling both sided constraints

$$L(x, \underline{z}, \overline{z}) = \frac{1}{2} x^T Q x + c^T x + \underline{z}^T (\underline{b} - Ax) + \overline{z}^T (Ax - \overline{b})$$

- Introduce slack variables $-\underline{b} \leq s \leq \underline{b}$

Handling both sided constraints

$$L(x, \underline{z}, \overline{z}) = \frac{1}{2} x^T Q x + c^T x + \underline{z}^T (\underline{b} - Ax) + \overline{z}^T (Ax - \overline{b})$$

- Introduce slack variables $-\underline{b} \leq s \leq \underline{b}$

Optimality Conditions

$$\begin{aligned} Qx + c - A^T \underline{z} + A^T \overline{z} &= 0 \\ Ax - s &= 0 \\ \underline{Z}\,(s - \underline{b}) &= 0 \\ \overline{Z}\,\left(\overline{b} - s\right) &= 0 \end{aligned}$$

Handling both sided constraints

$$L(x, \underline{z}, \overline{z}) = \frac{1}{2} x^T Q x + c^T x + \underline{z}^T (\underline{b} - Ax) + \overline{z}^T (Ax - \overline{b})$$

- Introduce slack variables $-\underline{b} \leq s \leq \underline{b}$

Optimality Conditions

$z_d := \overline{z} - \underline{z}$ and $z_a := \frac{\overline{z}+\underline{z}}{2}$

$$\begin{aligned} Qx + c - A^T \underline{z} + A^T \overline{z} &= 0 \\ Ax - s &= 0 \\ \underline{Z}\,(s - \underline{b}) &= 0 \\ \overline{Z}\left(\overline{b} - s\right) &= 0 \end{aligned} \qquad \begin{aligned} Qx + c + A^T z_d &= 0 \\ Ax - s &= 0 \\ (Z_a - Z_d/2)\,(s - \underline{b}) &= 0 \\ (Z_a + Z_d/2)\left(\overline{b} - s\right) &= 0 \end{aligned}$$

Handling both sided constraints

Newton's method + simplifications yield

$$\begin{pmatrix} Q & A^T \\ A & -\left(\overline{S}^{-1}\overline{Z} + \underline{S}^{-1}\underline{Z}\right)^{-1} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta z_d \end{pmatrix} =$$

$$\begin{pmatrix} -r_Q \\ -r_A + \left(\overline{S}^{-1}\overline{Z} + \underline{S}^{-1}\underline{Z}\right)^{-1} \left(-\underline{S}^{-1}\underline{r} + \overline{S}^{-1}\overline{r}\right) \end{pmatrix}$$

with

$$\Delta s = \left(\overline{S}^{-1}\overline{Z} + \underline{S}^{-1}\underline{Z}\right)^{-1} \left(-\underline{S}^{-1}\underline{r} + \overline{S}^{-1}\overline{r} + \Delta z_d\right)$$

$$\Delta \underline{z} = \underline{S}^{-1}\left(-\underline{r} - \underline{Z}\Delta s\right)$$

$$\Delta \overline{z} = \overline{S}^{-1}\left(-\overline{r} + \overline{Z}\Delta s\right).$$

Numerical Results

CUTEst Test Set

- Collection of continous optimization problems
- Version from May 8, 2018: In total 1305 problems

- Run WORHP using QPSOL and WORHP QP

Numerical Results

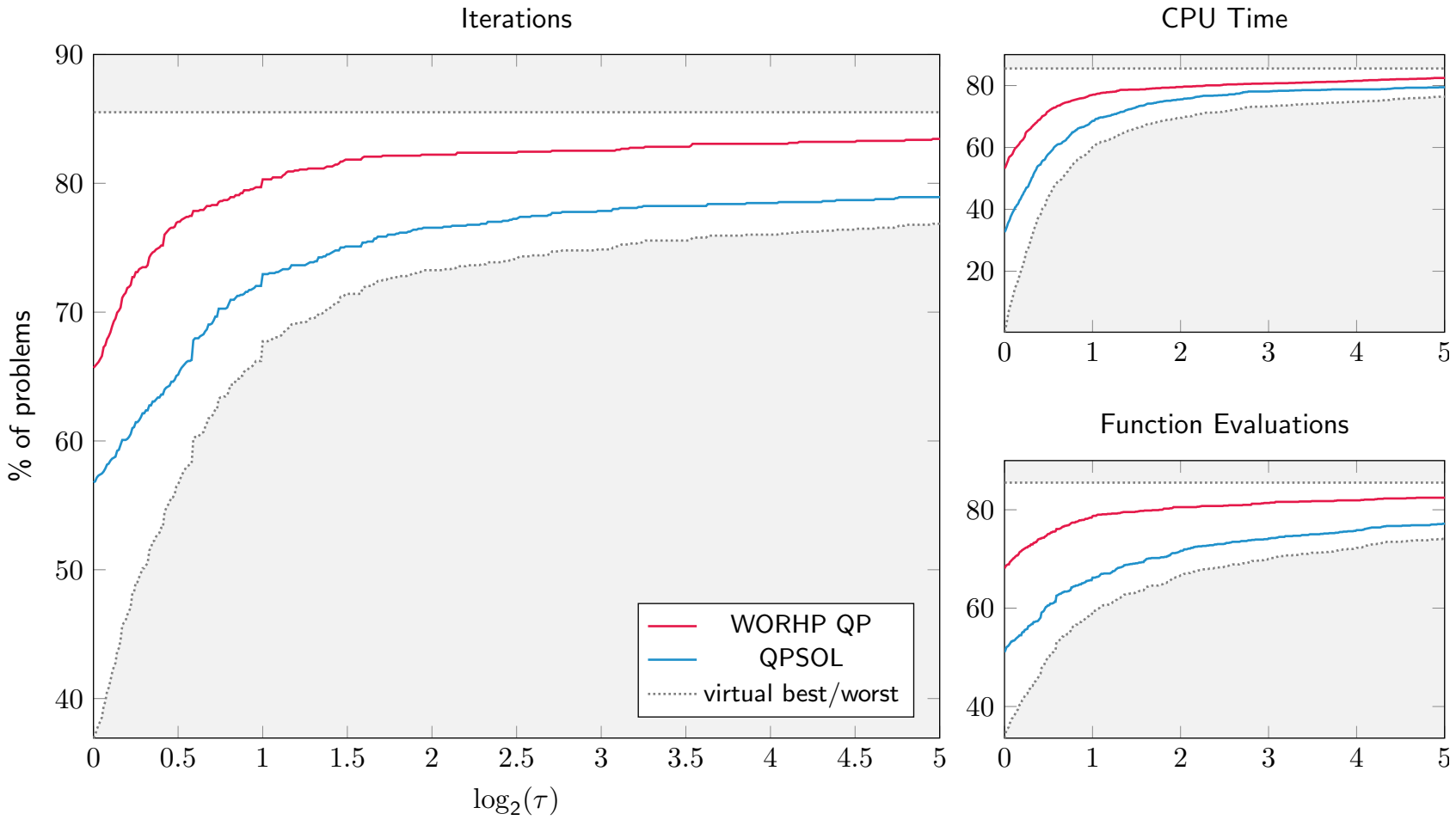
CUTEst Test Set

- Collection of continous optimization problems
- Version from May 8, 2018: In total 1305 problems

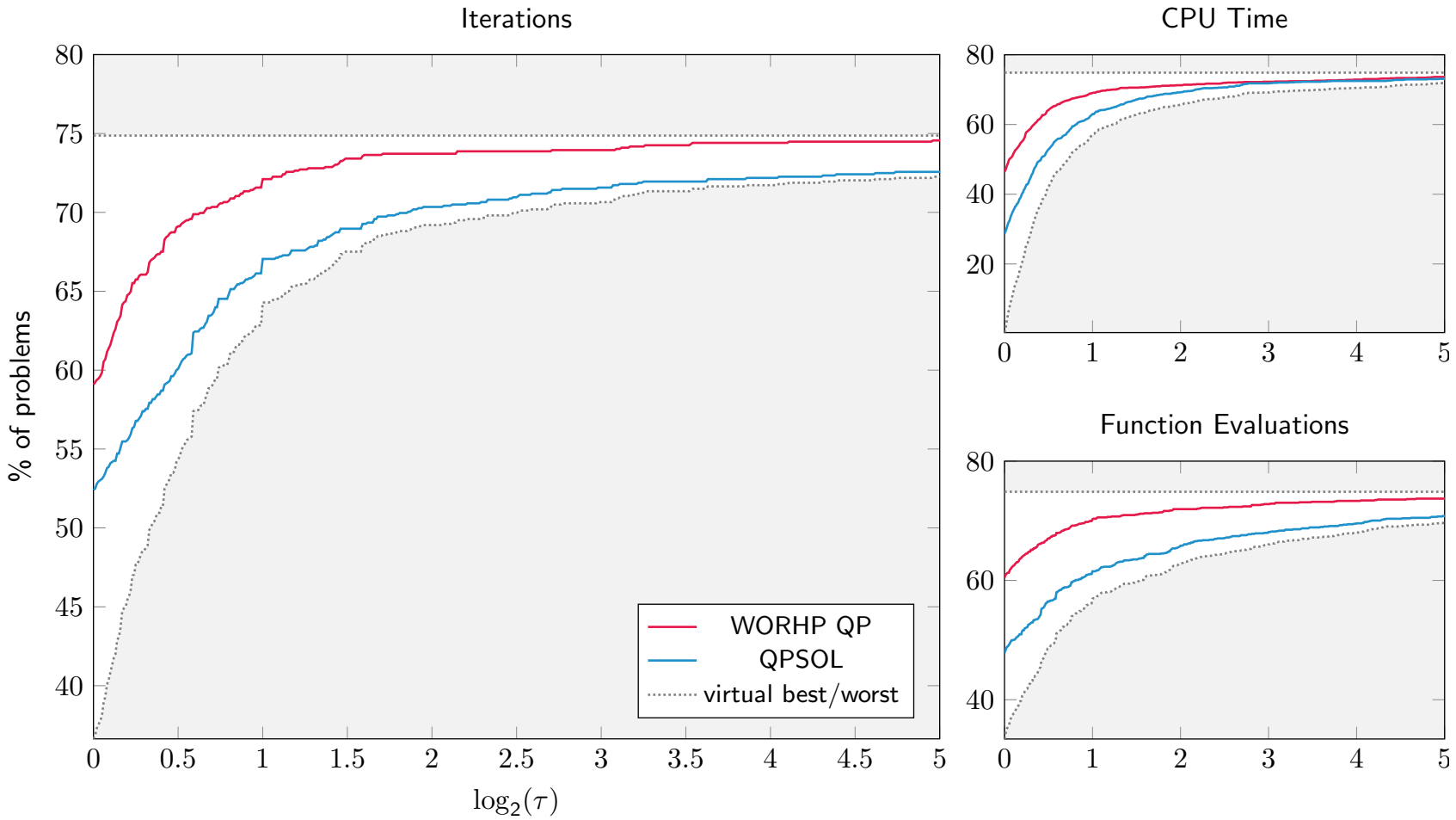
- Run WORHP using QPSOL and WORHP QP

Used QP solver	optimal	acceptable	unsuccessful
WORHP QP	1039	61	205
QPSOL	971	96	238

Performance Profiles



Performance Profiles (only equal minima)



Conclusion

- Implemented a reworked QP solver for WORHP
- Additional algorithmic features like Gondzio's multiple centrality correctors and weighted correctors
- Improved QP formulation for a cleaner interface between NLP and QP layer
- Greatly simplified handling of sensitivity derivatives
- Numerical results show significant improvements using the new solver

Conclusion

- Implemented a reworked QP solver for WORHP
- Additional algorithmic features like Gondzio's multiple centrality correctors and weighted correctors
- Improved QP formulation for a cleaner interface between NLP and QP layer
- Greatly simplified handling of sensitivity derivatives
- Numerical results show significant improvements using the new solver

- Code, stability and performance improvements for WORHP
- More versatile standalone QP solver as well