Derivative-Based Numerical Method for Penalty-Barrier Nonlinear Programming

Martin Peter Neuenhofen

October 26, 2018

Abstract

We present an NLP solver for nonlinear optimization with quadratic penalty terms and logarithmic barrier terms. The method is suitable for large sparse problems. Each iteration has a polynomial time-complexity. The method has global convergence and local quadratic convergence, with a convergence radius that depends little on our method but rather on the geometry of the problem.

Contents

1 Problem Statement **2**
- 1.1 Motivation of the Problem Statement . . . 2
- 1.2 Scope and Motivation of the Numerical Method . . . 2
- 1.3 Novel Mathematical Ideas of the Method . . . 3
- 1.4 Structure of the Paper . . . 4

2 Description of the Numerical Minimization Method **5**
- 2.1 Iterative Levels of Solution . . . 5
  - 2.1.1 Penalty-Barrier Problem . . . 5
  - 2.1.2 Modified Augmented Lagrangian Problem . . . 5
  - 2.1.3 Merit Model for the Computation of a Step Direction . . . 6
- 2.2 Globalization Technique . . . 7
  - 2.2.1 Line Search . . . 7
  - 2.2.2 Trust-Region Update . . . 8
- 2.3 Computation of the Step Direction . . . 9

3 Discussion and Outlook **13**

1 Problem Statement

We consider computing local solutions to:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad \phi(\mathbf{x}) := \quad & f(\mathbf{x}) + \frac{\rho_E}{2} \cdot \|\mathbf{x}\|_{\mathbf{S}}^2 \\ + \quad & \frac{1}{2\cdot\omega_E} \cdot \|c(\mathbf{x})\|_{\mathbf{W}}^2 \\ - \quad & \tau_E \cdot \mathbf{g}^{\mathsf{T}} \cdot \big(\log(\mathbf{x}-\mathbf{x}_{\mathsf{L}}) + \log(\mathbf{x}_{\mathsf{R}}-\mathbf{x})\big) \end{aligned} \tag{NLP}$$

Therein, $\rho_E, \omega_E, \tau_E > 0$ very small; $\mathbf{g} \in \mathbb{R}^n$ strictly positive, $\mathbf{S} \in \mathbb{R}^{n\times n}$ symmetric positive definite, $\mathbf{W} \in \mathbb{R}^{m\times m}$ diagonal positive definite, $\mathbf{x}_{\mathsf{L}} < \mathbf{x}_{\mathsf{R}} \in \mathbb{R}^n$; twice continuously differentiable functions

$$f : \mathbb{R}^n \to \mathbb{R}\,, \tag{1}$$

$$c : \mathbb{R}^n \to \mathbb{R}^m\,. \tag{2}$$

The Lagrangian function $\mathcal{L}(\mathbf{x},\boldsymbol{\lambda}) := f(\mathbf{x}) - \boldsymbol{\lambda}^{\mathsf{T}} \cdot c(\mathbf{x})$. Initial guesses $\mathbf{x}_0 \in \mathbb{R}^n, \boldsymbol{\lambda}_0 \in \mathbb{R}^m$, where $\mathbf{x}_{\mathsf{L}} < \mathbf{x}_0 < \mathbf{x}_{\mathsf{R}}$.

1.1 Motivation of the Problem Statement

Typically, one would formulate a problem like

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & c(\mathbf{x}) = \mathbf{0}\,, \qquad \mathbf{x}_{\mathsf{L}} \le \mathbf{x} \le \mathbf{x}_{\mathsf{R}}\,. \end{aligned} \tag{3}$$

The disadvantage of this problem over (NLP) is that there can be issues related to infeasibility (i.e. when there is no solution to $c(\mathbf{x}) = \mathbf{0}$), complementarity and constraint qualifications (i.e. when the optimality conditions have ill-posed Lagrange multipliers), as well as local uniqueness (i.e. when the local minimizer $\mathbf{x}$ is not strict). For sufficiently small values ρ_E (regularization of strictness), ω_E (regularization of possible collinearity of equality-constraints), τ_E (regularization of complementarity) these issues do not arise.

Further it is also advantageous that (NLP) makes sense when $m \ge n$, whereas the commonly used optimization packages IPOPT [19], KNITRO [4], WORHP [3] reject such problem statements as of 2018.

Beyond benefits of (NLP) with regard to regularity, there is the advantage that ϕ matches exactly with problem formats that originate from stable direct transcription methods for optimal control problems [17].

Below, we present a robust and locally efficient numerical method for computing stationary points $\mathbf{x}$.

1.2 Scope and Motivation of the Numerical Method

Nowadays, in the literature you find that NLP solvers are ranked by their performance when solving the CUTEr test set of optimization problems. That is

a set of mostly trivial test problems[1] whereas another fraction of 5% are just stated in a numerically untreatable way – indicating that this is not a problem collection made for systematically testing the robustness of solvers with regard to treatment of difficult problems.

In industry we strive for an entirely different goal, namely for methods that are reliable. It does not help if NLP solver XY is so fast that it can compute 10000 iterations per hour, when this means that after one hour it has performed indeed 10000 iterations but made zero progress, cf. to our experiments in [16]; or when it just does not converge for the particular problems that the user is interested in; or terminates with unsatisfactory criteria such as "restoration found point that is unacceptable to the filter", especially when it were initialized from a feasible point $\mathbf{x}_0$; or when a rescaling of the cost-function with a factor of 10 breaks the method.

In this work, we do not try to find just another milder globalization principle and barrier-parameter update heuristic for the line-search Newton-method on the ε-KKT equations [18]. Instead, we merge concepts from different types of methods to combine their strengths and arrive at a superior algorithm. In achieving this goal, we neglect concerns related to cost of the method. That is, our method can be considered as a Sequential Quadratic Programming (SQP) method[2], where the subproblem is solved with an Interior-Point method (IPM). It is understoof that for such an SQP method a single iteration is considerably more expensive than for an IPM, since in the former a QP has to be solved in each iteration (costs about 40 linsolves) whereas in IPM only one linear system has to be solved per iteration (i.e., costs one linsolve). However, SQP methods like WORHP that indeed solve the QP via IPM are widely used and acknowledged in industry for their superior robustness, especially for badly scaled problems. So, cost is secondary as long as it is tractable.

1.3 Novel Mathematical Ideas of the Method

Our objective ϕ looks like the merit function in the Sequential Unconstrained Minimization Technique [7]. Following the idea in the reference, we minimize a function like ϕ but with geometrically decreasing parameters $\omega \searrow \omega_E,\ \tau \searrow \tau_E$. Since ϕ is smooth, a KKT-based Newton-method will converge locally quadratic with a radius of quadratic convergence that depends on the magnitude of ω, τ; hence crucial that these decrease at a moderate rate. So far our method is pretty identical to the works of Fiacco and McCormick.

Their method decreases ω, τ in the same geometric rate (i.e. ω does not decrease faser than τ or vice versa). This can cause an issue because it is not clear how the ratio between ω, τ should be. For instance, when ω is too large then local convexity of ϕ may not be encountered because the constraints may add crucial convexity. Worse, for ω not sufficiently small the iterates may move away from feasible points and do not find back. As a first novelty, to get rid of

[1] Wächter and Biegler [19] notice that 75% of the problems can be solved with a naive Newton method on the ε-KKT system without any need for a globalization technique.

[2] though the subproblems are not exactly quadratic programs

this issue we use our modified augmented Lagrangian method [15] as a tailored treatment for the quadratic penalty terms with $c(\mathbf{x})$. This allows us to use relatively small penalty-parameters ω from the beginning and hence avoid the aforementioned issues related to convexity and feasibility.

When computing descent directions for minimizing the unconstrained objective, we avoid use of Newton steps. These steps rely on "centrality", meaning that only close to local minimizers the logarithmic terms of the objective are well-scaled. This is so because the Newton step is technically the minimizer of a quadratic model to the objective, cf. [5, 19, 9, 10]; and this quadratic approximation of the logarithmic barrier function is inefficient unless the current iterate is centralized. As a second novelty, by augmenting the quadratic model with logarithmic terms, we obtain descent directions that are superior over the Newton step. This is so simply because our log-augmented model gives a good fit to the objective even when the current iterate is not close to a local minimizer, because the log-terms are represented by the model in an exact way.

Finally, when it comes to incorporating a globlization technique for the step-length, an estimated amount of 95% of all proposed solvers and softwares on the internet use a line-search with the Armijo condition. While line-search methods lack behind trust-region methods by theoretical properties –in particular $\|\nabla^2_{\mathbf{x},\mathbf{x}}\mathcal{L}\|$ must be bounded at all iterates for line-search methods, while the same is not needed for trust-region [5]– , the Armijo condition in particular has little compatibility with the primal-dual merit functions [8] of interior-point methods. This is so because the strong convexity of the log-barriers causes convergence of $\|\nabla^2_{\mathbf{x},\mathbf{x}}\mathcal{L}\|$ to infinity as the barrier parameter τ decreases to zero, cf. the objective in [19, eqn. (3a)]. This illness is not easily repaired, because second-derivative information would be needed to account for highly convex features in the merit-function. But at the same time, when augmenting the Armijo condition with a quadratic Taylor term, the criterion would no longer guarantee a sufficient descent (in fact, not even a descent at all). As a third novelty, we therefore use a step-acceptance criterion that uses a quotient of predicted and actual descent (like in trust-region methods). The key to making this quotient insensitive to the current barrier parameter τ is that in our new method we compute the predicted decrease from our model that represents the log-terms in an exact way. Hence, the quotient does not converge to zero (which would imply small step-sizes) when the iterate is close to the border of the feasible region. Hence, we expect the method to not slow down (in contrast to many interior-point methods) when the initial guess or intermediate iterates come close to the border of the feasible region.

1.4 Structure of the Paper

In the following section we only present the actual method. We do not give further motivations than already given above to individual decisions that relate to the algorithmic components.

As represented by the table of contents, we first state the hierarchical levels of simplifying problem (NLP) into a sequence of simpler problems. This leads over

a penalty-barrier problem into an augmented Lagrangian problem. A model for determining descent directions for solving the latter is obtained from a quasi-quadratic approximation of the augmented Lagrangian problem.

In a second part we introduce the globalization technique, that uses both a line-search and a trust-region principle. Eventually, in the third part we explain how to form a practical method for minimization of the merit model for computation of a step direction.

We close the paper with a discussion and outlook.

2 Description of the Numerical Minimization Method

2.1 Iterative Levels of Solution

2.1.1 Penalty-Barrier Problem

We replace solving (NLP) by solving successively for $(\omega, \tau) \searrow (\omega_E, \tau_E)$ the problem:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n, \boldsymbol{\lambda}\in\mathbb{R}^m} \quad \psi^{II}(\mathbf{x}, \boldsymbol{\lambda}) := \quad & f(\mathbf{x}) + \frac{\rho_E}{2} \cdot \|\mathbf{x}\|^2_{\mathbf{S}} \\ + \quad & \frac{1}{2\cdot\omega} \cdot \Big(\|\mathbf{W}\cdot c(\mathbf{x})\|^2_{\mathbf{W}^{-1}} + \|\mathbf{W}\cdot c(\mathbf{x}) + \omega\cdot\boldsymbol{\lambda}\|^2_{\mathbf{W}^{-1}} \Big) \\ - \quad & \tau \cdot \mathbf{g}^{\mathsf{T}} \cdot \big(\log(\mathbf{x} - \mathbf{x}_{\mathsf{L}}) + \log(\mathbf{x}_{\mathsf{R}} - \mathbf{x}) \big) \end{aligned} \tag{4}$$

We consider that problem solved for respective ω, τ when

$$F^{II}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}_{\mathsf{L}}, \boldsymbol{\mu}_{\mathsf{R}}) := \begin{bmatrix} \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) + \rho_E \cdot \mathbf{S}\cdot\mathbf{x} + \mathbf{G}\cdot(\boldsymbol{\mu}_{\mathsf{R}} - \boldsymbol{\mu}_{\mathsf{L}}) \\ \mathbf{W}\cdot c(\mathbf{x}) + \omega\cdot\boldsymbol{\lambda} \\ \boldsymbol{\mu}_{\mathsf{L}} \circ (\mathbf{x} - \mathbf{x}_{\mathsf{L}}) - \tau\cdot\mathbf{1} \\ \boldsymbol{\mu}_{\mathsf{R}} \circ (\mathbf{x}_{\mathsf{R}} - \mathbf{x}) - \tau\cdot\mathbf{1} \end{bmatrix} \tag{5}$$

is zero (i.e., in practice its norm is smaller than a tolerance). That is because in contrast to the residual of F^{II} the gradient of ψ^{II} is badly scaled and hence unsuitable for a convergence check.

2.1.2 Modified Augmented Lagrangian Problem

The penalty-barrier problem in turn is solved either directly (in later stages) or via a modified augmented Lagrangian method. In case of the latter, we keep $\boldsymbol{\lambda}$

fixed and solve:

$$\begin{aligned}\min_{\mathbf{x}\in\mathbb{R}^n,\mathbf{y}\in\mathbb{R}^m} \quad \psi^I(\mathbf{x},\mathbf{y}) := \quad & \mathcal{L}(\mathbf{x},\boldsymbol{\lambda}) + \frac{\rho_E}{2}\cdot\|\mathbf{x}\|^2_{\mathbf{S}} \\ + \quad & \frac{1}{2}\cdot\Big\|\mathbf{W}\cdot c(\mathbf{x}) + \omega\cdot\boldsymbol{\lambda}\Big\|^2_{\mathbf{W}^{-1}\cdot(\omega\cdot\mathbf{I}+\mathbf{D})^{-1}} \\ + \quad & \frac{1}{2}\cdot\Big\|\mathbf{W}\cdot c(\mathbf{x}) + \omega\cdot\boldsymbol{\lambda} + (\omega\cdot\mathbf{I}+\mathbf{D})\cdot\mathbf{y}\Big\|^2_{\mathbf{W}^{-1}\cdot(\omega\cdot\mathbf{I}+\mathbf{D})^{-1}} \\ - \quad & \tau\cdot\mathbf{g}^{\mathsf{T}}\cdot\big(\log(\mathbf{x}-\mathbf{x}_{\mathsf{L}}) + \log(\mathbf{x}_{\mathsf{R}}-\mathbf{x})\big)\end{aligned} \tag{MALM}$$

We consider that problem solved for respective $\omega, \tau, \boldsymbol{\lambda}$ when

$$F^I(\mathbf{x},\mathbf{y},\boldsymbol{\mu}_{\mathsf{L}},\boldsymbol{\mu}_{\mathsf{R}}) := \begin{bmatrix}\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x},\boldsymbol{\lambda}+\mathbf{y}) + \rho_E\cdot\mathbf{S}\cdot\mathbf{x} + \mathbf{G}\cdot(\boldsymbol{\mu}_{\mathsf{R}}-\boldsymbol{\mu}_{\mathsf{L}}) \\ \mathbf{W}\cdot c(\mathbf{x}) + \omega\cdot\boldsymbol{\lambda} + (\omega\cdot\mathbf{I}+\mathbf{D})\cdot\mathbf{y} \\ \boldsymbol{\mu}_{\mathsf{L}}\circ(\mathbf{x}-\mathbf{x}_{\mathsf{L}}) - \tau\cdot\mathbf{1} \\ \boldsymbol{\mu}_{\mathsf{R}}\circ(\mathbf{x}_{\mathsf{R}}-\mathbf{x}) - \tau\cdot\mathbf{1}\end{bmatrix} \tag{6}$$

is zero (i.e., in practice its norm is smaller than a tolerance). $\mathbf{D},\mathbf{W}\in\mathbb{R}^{m\times m}$ must be diagonal positive definite. Once solved, we update:

$$\boldsymbol{\lambda} := \boldsymbol{\lambda} + \mathbf{y} \tag{7}$$

Then, we minimize ψ^I for the new parameters $\omega, \tau, \boldsymbol{\lambda}$. Always using initial guess $\mathbf{y}_0 := \mathbf{0}$, we repeat minimization of $\phi^{I,mod}$ with the updated $\boldsymbol{\lambda}$ until $\mathbf{y}\to\mathbf{0}$.

From a linearised analysis we find that in the limit the norm of the minimizers $\mathbf{y}$ converges to zero like

$$\|\mathbf{y}_{k+1}\| \le \|\mathbf{W}\cdot(\mathbf{W}+\omega\cdot\mathbf{I}+\mathbf{D})^{-1}\|\cdot\|\mathbf{y}_k\| \tag{8}$$

where k is the iterator over the number of updates of $\boldsymbol{\lambda}$. Thus, choosing $\mathbf{D}$ appropriate (e.g. $\mathbf{D} = 10\cdot\omega\cdot\mathbf{W}$) yields reasonably fast convergence of $\mathbf{y}$ and hence of $\boldsymbol{\lambda}$.

2.1.3 Merit Model for the Computation of a Step Direction

We find descent directions to ψ^I by minimizing the following model of ψ^I.

$$\begin{aligned}\psi^{I,mod}(\mathbf{d}_{\mathbf{x}},\mathbf{d}_{\mathbf{y}}) := \quad & \frac{1}{2}\cdot\mathbf{d}_{\mathbf{x}}^{\mathsf{T}}\cdot\Big(\nabla^2_{\mathbf{x}}\mathcal{L}(\mathbf{x},\boldsymbol{\lambda}+\mathbf{y}) + \rho\cdot\mathbf{S}\Big)\cdot\mathbf{d}_{\mathbf{x}} + \nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x},\boldsymbol{\lambda}+\mathbf{y})^{\mathsf{T}}\cdot\mathbf{d}_{\mathbf{x}} \\ + \quad & \frac{1}{2}\cdot\Big\|\mathbf{W}\cdot\big(c(\mathbf{x}) + \nabla c(\mathbf{x})^{\mathsf{T}}\cdot\mathbf{d}_{\mathbf{x}}\big) + \omega\cdot\boldsymbol{\lambda}\Big\|^2_{\mathbf{W}^{-1}\cdot\hat{\mathbf{D}}^{-1}} \\ + \quad & \frac{1}{2}\cdot\Big\|\mathbf{W}\cdot\big(c(\mathbf{x}) + \nabla c(\mathbf{x})^{\mathsf{T}}\cdot\mathbf{d}_{\mathbf{x}}\big) + \omega\cdot\boldsymbol{\lambda} + \hat{\mathbf{D}}\cdot(\mathbf{y}+\mathbf{d}_{\mathbf{y}})\Big\|^2_{\mathbf{W}^{-1}\cdot\hat{\mathbf{D}}^{-1}} \\ - \quad & \tau_F\cdot\mathbf{g}^{\mathsf{T}}\cdot\big(\log(\mathbf{x}+\mathbf{d}_{\mathbf{x}}-\mathbf{x}_{\mathsf{L}}) + \log(\mathbf{x}_{\mathsf{R}}-\mathbf{x}-\mathbf{d}_{\mathbf{x}})\big)\end{aligned} \tag{TRQP}$$

Therein, $\tau_F := \tau$, the current barrier parameter, and $\hat{\mathbf{D}} := \omega\cdot\mathbf{I}+\mathbf{D}$. That model is minimized subject to

$$\|\mathbf{d}_{\mathbf{x}}\|_\infty \le \Delta_{\mathbf{x}} \tag{9}$$

$$\|\mathbf{d}_{\mathbf{y}}\|_\infty \le \Delta_{\mathbf{y}} \tag{10}$$

Choosing $\rho \geq \rho_E$ sufficiently large, the merit model can be globally minimized for the unique minimizers $\mathbf{d_x} \in \mathbb{R}^n, \mathbf{d_y} \in \mathbb{R}^m$ with interior-point methods for convex quadratic programming. In practice, the computational cost for that is about equivalent to solving 40 linear equation systems. Details on solving that problem are given in a subsection below.

When minimizing $\psi^{I,mod}$, we get values for $\boldsymbol{\mu}_\mathsf{L}, \boldsymbol{\mu}_\mathsf{R}$ for free. These give a match for F^I, whenever $\mathbf{x}, \mathbf{y}$ converges to a root of it; i.e. we insert those Greeks into F^I (and F^{II}) to check whether the (respective) problem has been solved yet.

2.2 Globalization Technique

We have computed the step direction in a trust-region with radii $\Delta_\mathbf{x}, \Delta_\mathbf{y}$. We now perform a line-search along the found direction. Afterwards, we update $\Delta_\mathbf{x}, \Delta_\mathbf{y}$ in preparation for the subsequent iteration.

2.2.1 Line Search

Once the minimizer $\mathbf{d_x}, \mathbf{d_y}$ to $\psi^{I,mod}$ has been computed, we update $\mathbf{x}, \mathbf{y}$ as

$$\mathbf{x} := \mathbf{x} + \alpha_\mathbf{x} \cdot \mathbf{d_x} \tag{11}$$

$$\mathbf{y} := \mathbf{y} + \alpha_\mathbf{y} \cdot \mathbf{d_y}\,. \tag{12}$$

Primal and dual step lengths $\alpha_\mathbf{x}, \alpha_\mathbf{y} \in (0, 1]$ are computed according to a sufficient decrease condition[3].

To define the descent condition, we use the writings

$$\psi^I_\alpha(\alpha_\mathbf{x}, \alpha_\mathbf{y}) := \psi^I(\mathbf{x} + \alpha_\mathbf{x} \cdot \mathbf{d_x}, \mathbf{y} + \alpha_\mathbf{y} \cdot \mathbf{d_y}) \tag{13}$$

$$\psi^{I,mod}_\alpha(\alpha_\mathbf{x}, \alpha_\mathbf{y}) := \psi^{I,mod}(\alpha_\mathbf{x} \cdot \mathbf{d_x}, \alpha_\mathbf{y} \cdot \mathbf{d_y}) \tag{14}$$

and the *optimal dual step-length*

$$\alpha^{\text{opt}}_\mathbf{y}(\alpha_\mathbf{x}) := \underset{\alpha_\mathbf{y} \in (0,1]}{\arg\min} \left\{ \psi^I_\alpha(\alpha_\mathbf{x}, \alpha_\mathbf{y}) \right\}, \tag{15}$$

which can be computed analytically when $\alpha_\mathbf{x}$ is fixed; hence here written as function with argument $\alpha_\mathbf{x}$. The analytic formula for $\alpha^{\text{opt}}_\mathbf{y}$ is cheap.

We define the quotient of actual merit decrease and predicted merit decrease by the model[4]:

$$\eta(\alpha_\mathbf{x}, \alpha_\mathbf{y}) := \frac{\psi^I_\alpha(0,0) - \psi^I_\alpha(\alpha_\mathbf{x}, \alpha_\mathbf{y})}{\psi^{I,mod}_\alpha(0,0) - \psi^{I,mod}_\alpha(\alpha_\mathbf{x}, \alpha_\mathbf{y})} \tag{16}$$

[3] Notice the iterates remain strictly interior because the log-barriers have been incorporated into the model for the step direction.

[4] Notice that quotient converges to 1 as $\mathbf{x}$ moves to a border of the box defined by $\mathbf{x}_\mathsf{L}, \mathbf{x}_\mathsf{R}$; thus the quotient is well-defined and strict interiorness of an initial guess is actually not required when using the limit. Further notice that the denominator in η is strictly positive (since $\phi^{I,mod}$ is convex and globally minimized) or zero; the latter only when $\mathbf{x}$ is a critical point for ϕ^I; hence η gives a sufficient decrease condition.

We use the short writing

$$\eta(\alpha_{\mathbf{x}}) := \eta\big(\alpha_{\mathbf{x}}, \alpha_{\mathbf{y}}^{\mathrm{opt}}(\alpha_{\mathbf{x}})\big)\,. \tag{17}$$

The descent condition is as follows:

$$\eta(\alpha_{\mathbf{x}}) \geq 0.1 \tag{18}$$

That means the step length gives a significant decrease to ψ^I.

The algorithmic principle for computing $\alpha_{\mathbf{x}}, \alpha_{\mathbf{y}}$ is as follows.

1: $\alpha_{\mathbf{x}} := 1$
2: **while** $\eta(\alpha_{\mathbf{x}}) < 0.1$ **do**
3: $\quad \alpha_{\mathbf{x}} := 0.8 \cdot \alpha_{\mathbf{x}}$
4: **end while**
5: $\breve{\alpha}_{\mathbf{y}} := \alpha_{\mathbf{y}}^{\mathrm{opt}}(\alpha_{\mathbf{x}})$
6: $\alpha_{\mathbf{y}} := 1$
7: **while** $\eta(\alpha_{\mathbf{x}}, \alpha_{\mathbf{y}}) < 0.1$ **and** $\alpha_{\mathbf{y}} > \breve{\alpha}_{\mathbf{y}}$ **do**
8: $\quad \alpha_{\mathbf{y}} := \max\{\, \breve{\alpha}_{\mathbf{y}}\,,\, 0.8 \cdot \alpha_{\mathbf{y}} \,\}$
9: **end while**
10: **return** $\alpha_{\mathbf{x}}, \alpha_{\mathbf{y}}$

The goal of the line-search was to select $\alpha_{\mathbf{x}}, \alpha_{\mathbf{y}}$ largest possible, as we consider this in general more advantageous than actually minimizing ψ^I along the line. Reason: $\mathbf{d}_{\mathbf{x}}, \mathbf{d}_{\mathbf{y}}$ are coupled and their coupling assumption in (TRQP) is $\alpha_{\mathbf{x}} = \alpha_{\mathbf{y}} = 1$. The trust-region framework is incorporated to just achieve this, by avoiding that small step-sizes are selected during several iterations in a row.

Our descent condition gives sufficient decrease, just like Armijo's condition. But, in contrast to the latter, our condition will allow larger step-sizes when the merit function is locally strongly convex, i.e. when the slope of merit in $\alpha = 0$ is way steeper than for $\alpha > 0$ (as is the case for logarithmic barrier functions whenever the current iterate lives close to the border of the feasible region).

2.2.2 Trust-Region Update

Trust-region methods are superior in convergence theory over line-search methods. So we marry the two.

For this paragraph, foot-index ξ is for both foot-index $\mathbf{x}, \mathbf{y}$; e.g., Δ_ξ, α_ξ.

Depending on the current iteration, trust-region radii $\Delta_{\mathbf{x}}, \Delta_{\mathbf{y}} > 0$ are selected for the subsequent iteration.

Increase/ Decrease Conditions Whenever $\alpha_\xi < 1$, then Δ_ξ must be decreased. Only if both $\alpha_{\mathbf{x}}, \alpha_{\mathbf{y}} = 1$ and $\eta(1,1) > 0.75$, then $\Delta_{\mathbf{x}}, \Delta_{\mathbf{y}}$ are increased. In summary, exactly one of the following does happen to the radii:

- Both $\Delta_{\mathbf{x}}, \Delta_{\mathbf{y}}$ increase. That is when $\alpha_{\mathbf{x}} = 1$ and $\alpha_{\mathbf{y}} = 1$ and $\eta(\alpha_{\mathbf{x}}, \alpha_{\mathbf{y}}) > 0.75$.
- A Δ_ξ decreases. That is when $\alpha_\xi < 1$.

The second item can mean that zero, one, or two trust-region radii are decreased.

Increase/ Decrease Rule We use the following rule for increase:

$$\Delta_\xi := \max\Big\{\ \Delta_\xi\ ,\ \min\big\{\ 3\cdot\Delta_\xi\ ,\ 10\cdot\|\alpha_\xi\cdot\mathbf{d}_\xi\|_\infty\ \big\}\ \Big\} \tag{19}$$

We use the following rule for decrease:

$$\Delta_\xi := \min\big\{\ 0.5\cdot\Delta_\xi\ ,\ 10\cdot\|\alpha_\xi\cdot\mathbf{d}_\xi\|_\infty\ \big\} \tag{20}$$

The reason for choosing the rules like that is as follows:

In the beginning, the radii are often too large. Steps $\|\alpha_\xi\cdot\mathbf{d}_\xi\|_\infty$ are made that may be several orders of magnitude smaller than the radius. Rapid adaption is possible because then the second argument in the decrease formula is smaller.

In the end, unit steps will be accepted, due to local quadratic convergence. Then, $\|\alpha_\xi\cdot\mathbf{d}_\xi\|_\infty$ becomes small, but an increase of trust-region radius is triggered. The increase formula avoids that this trigger for increase will yield an actual increase in value of Δ_ξ whenever the steps are very small (which typically happens in final iterations of quadratic convergence).

The factor 10 in front of $\|\alpha_\xi\cdot\mathbf{d}_\xi\|_\infty$ means that we do not attempt using the current step size as an indicator of the subsequent one. We give the method some slack, by an order of magnitude, so the eventual SQP iterates originate from sub-problems whose trust-region constraints are inactive. That makes sure we do not destroy second-order local convergence.

2.3 Computation of the Step Direction

We now explain how the step directions can be computed, i.e. how $\psi^{I,mod}$ can be minimized in the primal-dual box. To avoid clutter, we write $\mathbf{x},\mathbf{y}$ instead of $\mathbf{d_x},\mathbf{d_y}$, and $\mathbf{D}$ instead of $\hat{\mathbf{D}}$. Also, τ may be used here free from context above (i.e. in this subsection we are in a new name-space of τ).

In fact, we identify the problem as minimizing, for $\tau\searrow\tau_F$ and $\pi\searrow+0$, the function:

$$\begin{aligned} w_{\tau,\pi}(\mathbf{x},\mathbf{y}) := \quad & \frac{1}{2}\cdot\mathbf{x}^\mathsf{T}\cdot\Big(\mathbf{Q}+\rho\cdot\mathbf{S}\Big)\cdot\mathbf{x}+\mathbf{c}^\mathsf{T}\cdot\mathbf{x} \\ + \quad & \frac{1}{2}\cdot\Big\|\mathbf{W}\cdot\big(\mathbf{A}\cdot\mathbf{x}-\mathbf{b}\big)\Big\|^2_{\mathbf{W}^{-1}\cdot\mathbf{D}^{-1}} \\ + \quad & \frac{1}{2}\cdot\Big\|\mathbf{W}\cdot\big(\mathbf{A}\cdot\mathbf{x}-\mathbf{b}\big)+\mathbf{D}\cdot\mathbf{y}\Big\|^2_{\mathbf{W}^{-1}\cdot\mathbf{D}^{-1}} \\ - \quad & \tau\cdot\mathbf{g}^\mathsf{T}\cdot\big(\log(\mathbf{x}-\mathbf{x}_\mathsf{L})+\log(\mathbf{x}_\mathsf{R}-\mathbf{x})\big) \\ - \quad & \pi\cdot\mathbf{g}^\mathsf{T}\cdot\big(\log(\Delta_\mathbf{x}\cdot\mathbf{1}+\mathbf{x})+\log(\Delta_\mathbf{x}\cdot\mathbf{1}-\mathbf{x})\big) \\ - \quad & \pi\cdot\mathbf{w}^\mathsf{T}\cdot\big(\log(\Delta_\mathbf{y}\cdot\mathbf{1}+\mathbf{y})+\log(\Delta_\mathbf{y}\cdot\mathbf{1}-\mathbf{y})\big) \end{aligned} \tag{21}$$

The matrices and vectors are just defined suitable as the Hessian, Jacobian, and so on, according to $\phi^{I,mod}$.

The feasible region for $\mathbf{y}$ is the open box centred in origin with diameter $2 \cdot \Delta_{\mathbf{y}}$. The feasible region for $\mathbf{x}$ is Ω_Δ:

$$\mathcal{B} := \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}_\mathsf{L} < \mathbf{x} < \mathbf{x}_\mathsf{R}\} \tag{22}$$

$$\mathcal{S}_\Delta := \left\{ \boldsymbol{\xi} \in \mathbb{R}^n \;\middle|\; -\Delta_{\mathbf{x}} \cdot \mathbf{1} < \mathbf{x} < \Delta_{\mathbf{x}} \cdot \mathbf{1} \right\} \tag{23}$$

$$\Omega_\Delta := \mathcal{B} \cap \mathcal{S}_\Delta\,. \tag{24}$$

The KKT conditions are, where $\mathbf{u} \circ \mathbf{v} := \mathit{diag}(\mathbf{u}) \cdot \mathbf{v}$:

$$\mathbf{0} = (\mathbf{Q} + \rho \cdot \mathbf{S}) \cdot \mathbf{x} + \mathbf{c} - \mathbf{A}^\mathsf{T} \cdot (\boldsymbol{\lambda}_\mathbf{x} + \boldsymbol{\lambda}_\mathbf{y}) + \mathbf{G} \cdot (\boldsymbol{\mu}_\mathsf{R} - \boldsymbol{\mu}_\mathsf{L} + \boldsymbol{\nu}^\mathbf{x}_\mathsf{R} - \boldsymbol{\nu}^\mathbf{x}_\mathsf{L}) \tag{25a}$$

$$\mathbf{0} = -\mathbf{W}^{-1} \cdot \mathbf{D} \cdot \boldsymbol{\lambda}_\mathbf{y} + \mathbf{W} \cdot (\boldsymbol{\nu}^\mathbf{y}_\mathsf{R} - \boldsymbol{\nu}^\mathbf{y}_\mathsf{L}) \tag{25b}$$

$$\mathbf{0} = \mathbf{W} \cdot (\mathbf{A} \cdot \mathbf{x} - \mathbf{b}) + \mathbf{D} \cdot \boldsymbol{\lambda}_\mathbf{x} \tag{25c}$$

$$\mathbf{0} = \mathbf{W} \cdot (\mathbf{A} \cdot \mathbf{x} - \mathbf{b}) + \mathbf{D} \cdot \mathbf{y} + \mathbf{D} \cdot \boldsymbol{\lambda}_\mathbf{y} \tag{25d}$$

$$\mathbf{0} = \boldsymbol{\mu}_\mathsf{L} \circ (\mathbf{x} - \mathbf{x}_\mathsf{L}) - \tau_F \cdot \mathbf{1} \tag{25e}$$

$$\mathbf{0} = \boldsymbol{\mu}_\mathsf{R} \circ (\mathbf{x}_\mathsf{R} - \mathbf{x}) - \tau_F \cdot \mathbf{1} \tag{25f}$$

$$\mathbf{0} = \boldsymbol{\nu}^\mathbf{x}_\mathsf{L} \circ (\Delta_\mathbf{x} \cdot \mathbf{1} + \mathbf{x}) - \pi \cdot \mathbf{1} \tag{25g}$$

$$\mathbf{0} = \boldsymbol{\nu}^\mathbf{x}_\mathsf{R} \circ (\Delta_\mathbf{x} \cdot \mathbf{1} - \mathbf{x}) - \pi \cdot \mathbf{1} \tag{25h}$$

$$\mathbf{0} = \boldsymbol{\nu}^\mathbf{y}_\mathsf{L} \circ (\Delta_\mathbf{y} \cdot \mathbf{1} + \mathbf{y}) - \pi \cdot \mathbf{1} \tag{25i}$$

$$\mathbf{0} = \boldsymbol{\nu}^\mathbf{y}_\mathsf{R} \circ (\Delta_\mathbf{y} \cdot \mathbf{1} - \mathbf{y}) - \pi \cdot \mathbf{1} \tag{25j}$$

Just to sketch how they are derived: The Greek variables are substitutes for the latter eight equations. When inserted into the first two, we obtain the gradients of $w_{\tau,\pi}$ after $\mathbf{x}$ and $\mathbf{y}$, where in the latter we replaced the first term with an expression of $\boldsymbol{\lambda}_\mathbf{y}$.

The reason for stating the equations as (25) is that they can be written as primal-dual Newton system whose Jacobian matrix can be symmetrized easily. The Jacobian is shown in Figure 1. We see that the matrix becomes symmetric by multiplying the rows from the left with the following matrices in order:

$$\mathbf{I} \tag{26}$$

$$\mathbf{I} \tag{27}$$

$$-\mathbf{W}^{-1} \tag{28}$$

$$-\mathbf{G} \cdot \mathit{diag}(\boldsymbol{\mu}_\mathsf{L})^{-1} \tag{29}$$

$$-\mathbf{G} \cdot \mathit{diag}(\boldsymbol{\mu}_\mathsf{R})^{-1} \tag{30}$$

$$-\mathbf{G} \cdot \mathit{diag}(\boldsymbol{\nu}^\mathbf{x}_\mathsf{L})^{-1} \tag{31}$$

$$-\mathbf{G} \cdot \mathit{diag}(\boldsymbol{\nu}^\mathbf{x}_\mathsf{R})^{-1} \tag{32}$$

$$-\mathbf{W} \cdot \mathit{diag}(\boldsymbol{\nu}^\mathbf{y}_\mathsf{L})^{-1} \tag{33}$$

$$-\mathbf{W} \cdot \mathit{diag}(\boldsymbol{\nu}^\mathbf{y}_\mathsf{R})^{-1} \tag{34}$$

$$
\left[\begin{array}{cc|cccccccc}
\mathbf{Q}+\rho\cdot\mathbf{S} & & -\mathbf{A}^{\mathsf{T}} & -\mathbf{A}^{\mathsf{T}} & -\mathbf{G} & \mathbf{G} & -\mathbf{G} & \mathbf{G} & & \\
 & & & -\mathbf{W}^{-1}\cdot\mathbf{D} & & & & & -\mathbf{W} & \mathbf{W} \\
\hline
\mathbf{W}\cdot\mathbf{A} & & \mathbf{D} & & & & & & & \\
\mathbf{W}\cdot\mathbf{A} & \mathbf{D} & & \mathbf{D} & & & & & & \\
\boldsymbol{\mu}_{\mathsf{L}} & & & & \mathbf{x}-\mathbf{x}_{\mathsf{L}} & & & & & \\
-\boldsymbol{\mu}_{\mathsf{R}} & & & & & \mathbf{x}_{\mathsf{R}}-\mathbf{x} & & & & \\
\boldsymbol{\nu}^{\mathbf{x}}_{\mathsf{L}} & & & & & & \Delta_{\mathbf{x}}+\mathbf{x} & & & \\
-\boldsymbol{\nu}^{\mathbf{x}}_{\mathsf{R}} & & & & & & & \Delta_{\mathbf{x}}-\mathbf{x} & & \\
 & \boldsymbol{\nu}^{\mathbf{y}}_{\mathsf{L}} & & & & & & & \Delta_{\mathbf{y}}+\mathbf{y} & \\
 & -\boldsymbol{\nu}^{\mathbf{y}}_{\mathsf{R}} & & & & & & & & \Delta_{\mathbf{y}}-\mathbf{y}
\end{array}\right]
$$

Figure 1: Jacobian of the KKT system (25). We dropped $\mathit{diag}(\cdot)$ for vectors.

After symmetrization, the Newton system fits into the form

$$
\begin{bmatrix}\mathbf{H} & \mathbf{J}^{\mathsf{T}}\\ \mathbf{J} & -\mathbf{D}\end{bmatrix}\cdot\mathbf{d}=-\mathbf{r}\,. \tag{35}
$$

Selecting ρ We choose $\rho\geq\rho_E$ such that the problem is convex since only then we can be sure the computed KKT point is unique and the global minimizer of $\psi^{I,mod}$.

We choose $\tau_E\geq\pi_E$ and select $\rho\geq\rho_E$ so large that the following function ψ_τ is convex $\forall\mathbf{x}\in\Omega_\Delta$ when $\tau:=\tau_F/2$.

$$
\begin{aligned}
\psi_\tau(\mathbf{x}):= \quad & \frac{1}{2}\cdot\mathbf{x}^{\mathsf{T}}\cdot(\mathbf{Q}+\rho\cdot\mathbf{S})\cdot\mathbf{x}+\mathbf{c}^{\mathsf{T}}\cdot\mathbf{x}\\
&+\frac{1}{2}\cdot\|\mathbf{A}\cdot\mathbf{x}-\mathbf{b}\|^2_{\mathbf{W}\cdot\mathbf{D}^{-1}}\\
&-\tau\cdot\Big(\log(\mathbf{x}-\mathbf{x}_{\mathsf{L}})+\log(\mathbf{x}_{\mathsf{R}}-\mathbf{x})\Big)
\end{aligned} \tag{36}
$$

In [13] we explain why this is exactly the function that must be convex in the feasible domain of $\mathbf{x}$ in order to obtain efficiency for the path-following method. In fact, in the reference we show computational efficiency for a primal interior-point method, of which the method presented here can be considered the primal-dual variant.

ψ_τ is convex in Ω_Δ when

$$
\nabla^2\psi_{\tau_F}(\mathbf{x})\equiv\mathbf{H}:=\mathbf{Q}+\rho\cdot\mathbf{S}+\mathbf{A}^{\mathsf{T}}\cdot\mathbf{W}\cdot\mathbf{D}^{-1}\cdot\mathbf{A}+\frac{\tau_F}{2}\cdot\mathit{diag}\left(\frac{\mathbf{g}}{(\mathbf{x}-\mathbf{x}_{\mathsf{L}})^2}+\frac{\mathbf{g}}{(\mathbf{x}_{\mathsf{R}}-\mathbf{x})^2}\right) \tag{37}
$$

is positive definite $\forall\mathbf{x}\in\Omega_\Delta$. The worst-case candidate for $\mathbf{x}$ is $\mathbf{x}=\tilde{\mathbf{x}}_M$, where

$$
\tilde{\mathbf{x}}_M:=\max\Big\{-\Delta\cdot\mathbf{1},\min\{\Delta\cdot\mathbf{1},\mathbf{x}_M\}\Big\} \tag{38}
$$

$$
\mathbf{x}_M:=0.5\cdot(\mathbf{x}_{\mathsf{L}}+\mathbf{x}_{\mathsf{R}})\,. \tag{39}
$$

The min, max are meant in separate for each row of a respective vector. Example:

$$\min\left\{ \begin{bmatrix} -1 \\ -2 \\ 5 \end{bmatrix} \ , \ \begin{bmatrix} -2 \\ 3 \\ 18 \end{bmatrix} \right\} = \begin{bmatrix} -2 \\ -2 \\ 5 \end{bmatrix} .$$

For determination of ρ we use:

1: $\rho := \rho_E$, $\mathbf{x} := \mathbf{x}_M$
2: **while true do**
3: Assemble $\mathbf{H}$ from (37), using $\rho, \mathbf{x}$.
4: **if** $\mathbf{H}$ is positive definite **then**
5: **return** ρ
6: **else**
7: $\rho := 10 \cdot \rho$
8: **end if**
9: **end while**

Path-Following We explain how to solve (25), using a primal-dual long-step path-following method.

Initialization: We use central initial guesses for $\mathbf{x}_0, \mathbf{y}_0$. We choose $\tau = \pi$ sufficiently large, such that $\mathbf{x}_0, \mathbf{y}_0$ are minimizers of an associated primal barrier function. We then compute $\boldsymbol{\lambda}_\mathbf{x}, \boldsymbol{\lambda}_\mathbf{y}, ..., \boldsymbol{\nu}^\mathbf{y}_\mathsf{R}$ according to the latter eight equations. More to the initialization below.

Path-following for τ: We keep $\tau = \pi$ at same values. We use the Mehrotra predictor-corrector method [11] to solve (25) for decreasing values of $\tau = \pi$ until $\tau = \tau_F$.

Path-following for π: We keep $\tau := \tau_F$ fixed. We use the Mehrotra predictor-corrector method to solve (25) for decreasing values of π until $\pi \leq \chi \cdot \pi$, where $\chi = 10^{-3}$.

While for the long-step Mehrotra predictor-corrector method proposed here there is no proof of computational efficiency, we have been able to prove that a primal variant of the proposed method is indeed computationally efficient, in the sense that its time-complexity is bounded by a polynomial in the input length times a logarithmic factor on norms of the matrices and vectors in $w_{\tau,\pi}$.

Initialization We introduce the associated self-concordant [12] primal barrier-function:

$$\hat{w}_\tau(\mathbf{x}, \mathbf{y}) := \frac{1}{\tau} \cdot w_{\tau,\tau}(\mathbf{x}, \mathbf{y}) \,, \tag{40}$$

i.e. where in $w_{\tau,\pi}$ the latter of the parameters is $\pi = \tau$.

We select $\mathbf{y}_0 := \mathbf{0}$, and $\mathbf{x}_0$ as the minimizer of

$$\begin{aligned} \Gamma(\mathbf{x}) := \quad & - \mathbf{g}^\mathsf{T} \cdot \big(\log(\mathbf{x} - \mathbf{x}_\mathsf{L}) + \log(\mathbf{x}_\mathsf{R} - \mathbf{x}) \big) \\ & - \mathbf{g}^\mathsf{T} \cdot \big(\log(\Delta_\mathbf{x} \cdot \mathbf{1} + \mathbf{x}) + \log(\Delta_\mathbf{x} \cdot \mathbf{1} - \mathbf{x}) \big) \end{aligned} \tag{41}$$

That minimizer is easily computed numerically (and does not depend on $\mathbf{g}$). To this end, use Newton iteration for each component of $\mathbf{x}_0$ individually, noticing that the Hessian of Γ is a diagonal matrix and hence there are no intercomponential connections. A suitable guess to start the Newton iteration with is given by the centre of mass of $\overline{\Omega_\Delta}$.

In [13] we showed that in exact arithmetic the described Newton iteration converges to 128 digits accuracy in 70 iterations at most. In practice, we are at machine accuracy usually after 6 iterations, no matter what $\mathbf{x}_\mathsf{L}, \mathbf{x}_\mathsf{R}, \Delta_\mathbf{x}$.

To determine τ_0, i.e. the initial value of τ and π, we use the primal barrier function. Given $\mathbf{x}_0, \mathbf{y}_0$, we compute the Newton decrement

$$\vartheta(\tau) := \sqrt{\nabla \hat{w}_\tau(\mathbf{x}_0, \mathbf{y}_0)^\mathsf{T} \cdot \nabla^2 \hat{w}_\tau(\mathbf{x}_0, \mathbf{y}_0) \cdot \nabla \hat{w}_\tau(\mathbf{x}_0, \mathbf{y}_0)} \tag{42}$$

for geometrically ascending values of τ_0 until $\vartheta(\tau_0) < 0.25$ [2]. The criterion makes sure that for this τ_0 the Newton iteration from $\mathbf{x}_0, \mathbf{y}_0$ converges quadratically to a point on the central path for $\tau = \tau_0$ [14].

According to this strategy, τ_0 may be large. But as a trade-off, the initial guess will have excellent centrality, shown in "Stable Interior-Point Method for Convex Quadratic Programming with Strict Error Bounds" [14]. Hence, path-following will be able to rapidly decrease τ. Hence, in practice it does not matter how large τ is in the beginning, and only a few dozens of path-following iterations will be needed until convergence anyway.

1: $\tau_0 := 1$
2: **while** $\vartheta(\tau_0) \geq 0.25$ **do**
3: $\quad \tau_0 := 10 \cdot \tau_0$
4: **end while**
5: **return** τ_0

After determining $\mathbf{x}_0, \mathbf{y}_0, \tau_0$, we compute the Greeks $\boldsymbol{\lambda}_\mathbf{x}, \boldsymbol{\lambda}_\mathbf{y}, \boldsymbol{\mu}_\mathsf{L}, \boldsymbol{\mu}_\mathsf{R}, \boldsymbol{\nu}^\mathbf{x}_\mathsf{L}, \boldsymbol{\nu}^\mathbf{x}_\mathsf{R}, \boldsymbol{\nu}^\mathbf{y}_\mathsf{L}, \boldsymbol{\nu}^\mathbf{y}_\mathsf{R}$ by inserting $\mathbf{x}_0, \mathbf{y}_0, \tau_0$ into (25c)–(25j), and solving these equations for the Greeks.

In summary, we have a starting point, and with this we begin the previously described path-following, first for $\tau = \pi$ decreasing and then for τ fixed and π further decreasing.

3 Discussion and Outlook

We discuss advantages and disadvantages of our method.

Advantages The method avoids some short-comings of current state-of-the-art methods. In particular:

For our method, the quality of the step-direction is not linked to a centrality-property of the current iterate. This is similar in advantage to as in SQP methods, where a concept like centrality is inherently avoided. In the same turn, we avoid the lack of second-order convergence, that most SQP methods suffer

from. That is because, as discussed more deeply in [13], our step directions have a convexization that arises from the log-barriers. Hence, in the limit, the shift ρ matches ρ_E and therefore the model merit function is a second-order accuarate approximation of the true merit function. In contrast to IPM, we consider the log-barriers exactly when forming a model for computing the descent direction, and so circumvent the major headache inherited in these methods when it comes to approximation quality of the log-barriers. In the same turn, we avoid the need for SQP to compute a suitable shift or an estimate to the active set, because we keep the log-barriers in the sub-problem and exploit their contribution to the convexity of the model of the merit function.

As a second benefit, we make use of Augmented Lagrangian techniques to treat large quadratic penalties/ equality constraints. These techniques are recognised for their robustness [6, 1]. Though only of linear rate of convergence, their convergence is fast and can be controlled by adapting the penalty parameters, as given by (8). The Augmented Lagrangian approach avoids Maratos-like effects[5] because the penalty parameter used for the merit function in the globalizaton technique remains of moderate size. Hence, second-order corrections are unnecessary. We notice that it would be difficult anyway to come up with a reasonable formula for a second-order correction in (NLP) because there is no such concept as a "feasible point"; that is, even the local minimizer of ϕ will usually not satisfy $c(\mathbf{x}) = \mathbf{0}$, and hence restoration of such equality would hinder convergence to the local minimizer.

Third and finally, we incorporated a novel sufficient-decrease criterion that computes the quotient of actual and predicted decrease. The significant advantage of our criterion lives in the fact that our model for the predicted decrease maintains the log-barrier terms in their original form. In consequence of this, the high curvature of the log-terms does not impede the step-length. Hence, the method is able to make good progress from decentralized points. Hence, not only the step direction but also the globalization technique of our method is robust with respect to the choice of the initial guess and with respect to centrality.

Disadvantages Our method is quite expensive, as we show by a crude count of the number of linear systems to be solved: We have all these levels of iterative solution. First, we have the decrease of ω, τ. Let's say this is a sequence of 8 problems. Within this sequence, we have a sequence of Augmented Lagrangian problems with a sequence of updates for $\boldsymbol{\lambda}$. Let's say this is a sequence of 5 problems, respectively. Thus, so far we have to solve 40 problems. Let's further say that each problem converges in 5 iterations (on average; in the limit it will be less, in the beginning it will be more). And that each iteration, solved via minimization of (21), requires about 40 linsolves on average. Then this makes

[5]Notice that Maratos' effect actually only arises for nonsmooth merit-functions, which we avoid. But for smooth functions it can still happen that a similar effect holds until one is very very close (e.g., at 10^{-30} distance) to the actual minimizer.

up to 8000 linear equation systems that may have to be solved. That is in hard contrast to maybe a couple of hundred iterations that an IPM may require to converge (if it converges, i.e. does not stall or terminate unsuccessful).

Outlook Thinking in relative terms, this is rather a worst-case calculation. In practice, through suitable adaptation of termination criteria for the subproblems and maybe more aggressive update rules (i.e. larger geometric fractions for decrease of ω, τ, more aggressive penalties for faster convergence of $\boldsymbol{\lambda}$) can bring a lot of change. Also, the subproblems are probably similar. Using warm-start techniques for the interior-point method for computing the step-direction as in [20], and using Krylov subspace recycling techniques for solving the related linear systems, it may be possible to decrease this cost by a huge fraction.

Further work could focus on the design of a method that tries to achieve similar convergence properties as the method presented while using suitable (more cheaply computable) approximations for solving the step-direction and globalization.

References

[1] E. Birgin, J. Martínez, and L. Prudente. Augmented Lagrangians with possible infeasibility and finite termination for global nonlinear programming. *Journal of Global Optimization*, 58(2):207–242, February 2014.

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[3] C. Büskens and D. Wassel. The ESA NLP Solver WORHP. In *Modeling and Optimization in Space Engineering*, pages 85–110. Springer New York, New York, NY, 2013.

[4] R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIOPT*, 9(4):877–900, 1999.

[5] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-Region Methods*. SIAM, Philadelphia, PA, USA, 2000.

[6] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Lancelot: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*. Springer Publishing Company, Incorporated, 1st edition, 2010.

[7] A.V. Fiacco and G.P. McCormick. *Nonlinear Programming: Sequential unconstrained minimization techniques*. John Wiley & Sons, 1968.

[8] A. Forsgren and P. E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. *SIOPT*, 8(4):1132–1152, 1998.

[9] A. Forsgren, P. E. Gill, and M. H. Wright. Interior methods for nonlinear optimization. *SIAM Rev.*, 44(4):525–597 (2003), 2002.

[10] M. Glavic and L. Wehenkel. Interior point methods: A survey, short survey of applications to power systems, and research opportunities. Technical report, Electrical Engineering and Computer Science Department, University of Liége, 2004.

[11] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIOPT*, 2(4):575–601, 1992.

[12] Y. Nesterov and A. Nemirovskii. *Interior-point polynomial algorithms in convex programming*, volume 13 of *SIAM Studies in Applied Mathematics*. SIAM, Philadelphia, PA, 1994.

[13] M. Neuenhofen. Weakly polynomial efficient minimization of a non-convex quadratic function with logarithmic barriers in a trust-region. *ArXiv e-prints*, June 2018.

[14] M. P. Neuenhofen. Stable interior-point method for convex quadratic programming with strict error bounds. *ArXiv e-prints. DOI:1711.01418*, 2017.

[15] M. P. Neuenhofen. Modified augmented Lagrangian method for the minimization of functions with quadratic penalty terms. *ArXiv e-prints. DOI:1804.08072*, 2018.

[16] M. P. Neuenhofen. PPD-IPM: Outer primal, inner primal-dual interior-point method for nonlinear programming. *ArXiv e-prints. DOI:1803.01829*, 2018.

[17] M. P. Neuenhofen and E. C. Kerrigan. Dynamic Optimization with Convergence Guarantees. *ArXiv e-prints*, October 2018.

[18] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.

[19] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.*, 106(1, Ser. A):25–57, 2006.

[20] E. A. Yildirim and S. J. Wright. Warm-start strategies in interior-point methods for linear programming. *SIOPT*, 12(3):782–810, 2002.