A one phase IPM for non-convex optimization

Oliver Hinder, Yinyu Ye

June 21, 2017

Abstract

Based on the work of [REF] it has been assumed that infeasible start IPM developed for conic optimization [REF] cannot be adapted to non-linear optimization and converge without significant modification i.e. a two phase method or a penalty function. We show that, in fact, by careful initialization and non-linear updates on the primal variables.

1 Introduction

[?] showed that if we apply an infeasible start IPM to:

$$\begin{aligned} &\min x \\ x^2 - s_1 - 1 &= 0 \\ x - s_2 - 1/2 &= 0 \\ s_1, s_2 &\geq 0 \end{aligned}$$

Fails to converge to either a local optimum or infeasibility certificate
Modifications:

A. Two phases (IPOPT)

B. Compute two directions (KNITRO)

C. Penalty (or big-M) method e.g. [?, ?]

1.1 Relevant literature

Within non-convex optimization there are four papers that I think are particularly relevant to our work:

A. The paper [?] shows that there are examples for which infeasible start algorithms will always fail to converge to either a optimal solution or a stationary measure of infeasibility when constraints are non-convex (irrespective of the strategy for used). This is the inspiration for the two phase algorithm of IPOPT and justifies why our one phase algorithm is necessary.

B. The description of the IPOPT algorithm [?]. IPOPT uses a two phase method the primary phase searches simultaneously for optimality and feasibility using a classical infeasible start method and a feasibility restoration phase that minimizes infeasibility. The feasibility restoration phase is only called when the step size for the infeasible start method is small. Another distinct feature of the algorithm is the filter line search (which allows progress on either the constraints or the objective).

C. The description of the KNITRO algorithm [?]. KNITRO is a trust region algorithm. The approach is quite distinct from typical infeasible start algorithms and is worth looking at (each step computes two different directions, using two different linear systems, one to reduce the objective and the other to reduce infeasibility). There is a more recent paper [?] that adds an feasibility restoration phase (this is theoretically unnecessary, but the practical results are good).

D. The paper [?] introduces a barrier penalty method. This paper uses a similar approach to us. The main different with our approach is we treat λ as a dual variable, whereas in Curtis's paper λ is replaced by a penalty parameter that is updated in an ad hoc fashion.

E. Papers in convex optimization?

2 Simplified algorithm

We wish to solve the following problem:

$$
\begin{aligned}
&\min f(x) \\
&a(x) \leq 0
\end{aligned}
$$

Assume:

A. The constraints and objective are C^2 - L_0, L_1 etc

B. The set $a(x) \leq \theta$ is bounded for any $\theta \in \mathbb{R}^m$

2.1 Infeasibility certificates

First-order local L1-infeasibility certificate, if x^* is a first-order local optimum of:

$$
\begin{aligned}
\min e^T z& \\
a(x) - s + z = 0& \\
s, z \geq 0&
\end{aligned}
$$

With $e^T z > 0$.

First-order local farkas infeasibility certificate, if there exists some $w \leq 0$, such that x^* is a first-order local optimum of:

$$\min w^T a(x)$$

With $w^T a(x) > 0$. Now, suppose we have found a point such that

$$\|w^T \nabla a(x)\|_2 < \frac{w^T a(x)}{\|x - x^*\|_2}$$

Now, if the function $w^T a(x)$ is convex then we have:

$$w^T a(x^*) \geq w^T a(x) + w^T \nabla a(x)(x - x^*) > 0$$

Therefore we declare a problem locally infeasible if our algorithm finds a point with:

$$\frac{\|w^T \nabla a(x)\|_2}{w^T a(x)} \leq \epsilon_{\mathbf{inf}}$$

Where $\epsilon_{\mathbf{inf}}$ is the tolerance for primal feasibility.

L_∞ ...

2.2 Derivation of method

Log barrier problem:

$$
\begin{aligned}
&\min f(x) - \mu^k \sum_i \log(s_i) \\
&a(x) + s = 0 \\
&s \geq 0
\end{aligned}
$$

Shifted log barrier problem

$$
\begin{aligned}
&\min f(x) - (1-\eta^k)\mu^k \sum_i \log s_i + \frac{\delta^k}{2}\|x - x^k\|^2 && (1) \\
&a(x) + s = (1-\eta^k)(a(x^k) + s^k) && (2) \\
&s \geq 0 && (3)
\end{aligned}
$$

KKT system:

$$\nabla f(x) + \delta^k (x - x^k) + y^T \nabla a(x) = 0$$
$$a(x) + s = (1 - \eta^k)(a(x^k) + s^k)$$
$$s_i y_i = (1 - \eta^k)\mu^k$$
$$s, y \geq 0$$

Corresponding newton system:

$$\begin{bmatrix} \nabla_x^2 L(x^k, y^k) + \delta^k I & \nabla a(x^k)^T & 0 \\ \nabla a(x^k) & 0 & I \\ 0 & S^k & Y^k \end{bmatrix} \begin{bmatrix} d_x^k \\ d_y^k \\ d_s^k \end{bmatrix} = \begin{bmatrix} -(\nabla f(x^k) + \nabla a(x^k)^T y^k) \\ -\eta^k (a(x^k) + s^k) \\ (1 - \eta^k)\mu^k e - Y^k s^k \end{bmatrix} \tag{4}$$

By taking the primal schur complement one can see solving system (4) is equivalent to solving:

$$(M(x^k, y^k, s^k) + \delta I) d_x^k = -\nabla f(x^k) - \nabla a(x^k)^T ((1 - \eta^k)\mu^k (S^k)^{-1} e + \eta^k Y^k (e + (S^k)^{-1}(a(x^k) + s^k)))$$

Where:

$$M(x, y, s) = \nabla^2 \mathcal{L}(x, y) + \nabla a(x)^T Y S^{-1} \nabla a(x) \tag{5}$$

$$D(x, y) \leq \mu^k \& \|Sy - \mu^k\|_\infty \leq \mu^k/4 \tag{6}$$

2.3 Algorithm behaviour

function SIMPLIFIED-ONE-PHASE-NON-CONVEX-IPM(x, y)
  for $k = 1, ...\infty$ **do**
    Stabilization step:
      Minimize the unconstrained problem (1) with $\eta^k = 0$ until (6) holds
    if $D(x, y) \leq \epsilon$ & $\mu^k \leq \epsilon$ **then**
      Terminate with optimal solution
    else if $\frac{\|\nabla a(x)^T y\|_2}{a(x)^T y} \leq \epsilon$ **then**
      Terminate at infeasible point
    end if
    if $D(x, y) \leq \mu^k$ **then**
      Aggressive correction:
        Solve system (4) with $\delta^k = \frac{\|g^k\|L_0}{\mu^k} - \lambda_{\min}(M^k)$ and $\eta^k = 1$
        $x^+, y^+, s^+, \alpha^+ \leftarrow$ AGGRESSIVE-LINE-SEARCH(f, a, x, y, d_x, d_y)
        if $\mu\alpha^+ < \min s_i^k/10$ **then**
          increase δ
        end if
    end if
  end for
end function

Algorithm 1 Aggressive line search

```
function AGGRESSIVE-LINE-SEARCH(f, a, x, y, d_x, d_y)
    η ← 1
    α_P ← FRACTIONTOBOUNDARY(s, d_s)
    for i = 1, ..., ∞ do
        x⁺, y⁺, status ← MOVE(f, a, x, y, d_x, d_y, η, α_P)
        if status = feasible & Function value does not increase too much then
            return x⁺, y⁺, s⁺
        else
            α_P ← α_P/2
        end if
    end for
end function
```

2.4 Behavior of this algorithm on convex quadratic programs

[one stabilization step followed by one aggressive step]. [resemblance to predictor corrector]

2.5 Convergence proofs

Lemma 1. *Assume that $a(x^1) - s^1 = -\mu e$ and $\epsilon < 1$. If the criterion for an aggressive step is met at iteration k then we have:*

$$\|y^k\|_1 \geq \frac{\|\nabla c(x^k)\|_2}{\epsilon^2} + 3m$$

Proof. Observe that:

$$-a(x)^T y = -(a(x) - s)^T y - s^T y \geq \mu(e^T y - 2)$$

Therefore:

$$\frac{\|\nabla a(x)^T y\|}{-a(x)^T y} \leq \frac{\mu^k \sqrt{\|y\|_1 + 1} + \|\nabla c(x)\|}{\mu(\|y\|_1 - 2m)}$$

If:

$$\|y^k\|_1 \geq \frac{\|\nabla c(x^k)\|_2 + 3m}{\epsilon^2}$$

Then:

$$\frac{\|\nabla a(x)^T y\|}{-a(x)^T y} \leq \epsilon$$

Which gives the result. □

Proof. Observe that:

$$-a(x)^T y = -(a(x) - s)^T y - s^T y \geq \mu(e^T y - 2)$$

Therefore:

$$\frac{\|\nabla a(x)^T y\|}{-a(x)^T y} \leq \frac{1 + \|\nabla c(x)\|}{\mu(\|y\|_1 - 2m)}$$

If:

$$\|y^k\|_1 \geq \frac{\|\nabla c(x^k)\|_2}{\epsilon^2} + 3m$$

Then:

$$\frac{\|\nabla a(x)^T y\|}{-a(x)^T y} \leq \epsilon$$

Which gives the result. □

Lemma 2. *Assume that $a(x^1) - s^1 = -\mu e$ and $\epsilon < 1$. The algorithm,* SIMPLIFIED-ONE-PHASE-NON-CONVEX-IPM*, with any takes at most $\frac{\mu^0(2\|\nabla c(x^k)\|_2 + 8)}{\epsilon^2}$ aggressive steps to terminate with either a first-order stationary point or first-order farkas certificate.*

Proof. We wish to prove that for any δ with

$$\delta \geq \frac{\|g^k\| L_0}{\mu^k} - \lambda_{\min}(M^k)$$

and α satisfying

$$\alpha \leq \frac{1}{\|y^k\|_\infty + 4} \tag{7}$$

the iterate $x^+ = x^k + \alpha d_x^k$, $y^+ = y^k + \alpha d_y^k$, $\mu^+ = \mu(1-\alpha)$ is feasible. Observe that this implies the result since if: $\alpha \geq \frac{1}{2(\|y^k\|_\infty + 4)}$ then:

$$\mu^{k+1} = (1-\alpha)\mu^k = \mu^k - \frac{\mu^k}{2\|y^k\|_\infty + 8} \leq \mu^k - \frac{\epsilon^2}{2\|\nabla c(x^k)\|_2 + 8}.$$

We wish to show that $s^{k+1} \in [s^k/2, 3s^k/2]$. Where $s^{k+1} = a(x + \alpha_P d_x) + (1 - \alpha_P)\mu^k e$. Subtracting and adding $s^k = a(x^k) + \mu^k e$ yields

$$s^{k+1} = s^k + (a(x^k + \alpha_P^k d_x^k) - a(x^k)) - \alpha_P^k \mu^k e$$

Therefore, it remains to bound the term $a(x^k + \alpha_P^k d_x^k) - a(x^k) - \alpha_P^k \mu^k e$. Applying our assumption on α^k, we immediately get $0 \leq \alpha_P^k \mu^k e \leq s^k/4$. Furthermore, we know that $\|d_x^k\|_2 \leq \mu^k L_0$ therefore:

$$\alpha_P^k \|d_x^k\|_2 \leq \frac{\min_i\{s_i^k\}}{2L_0}$$

Since $a(x)$ is L_0-Lipshitz we have:

$$-s^k/4 \leq a(x^k) - a(x^k + \alpha_P^k d_x^k) \leq s^k/4$$

which shows that $s^{k+1} \in [s^k/2, 2s^k]$. Observe that $y^{k+1} = y^k + \alpha^k d_y^k \geq y^k/2$. It remains to show that $\|y^{k+1}s^{k+1} - \mu^{k+1}\|_\infty \leq \mu^k/2$. Now we have:

$$d_y = -Y(S^{-1}d_s + e)$$

Hence using that $\|d_s\| \leq ...$ we get $d_y \in [-2y, 2y]$. It follows that $y^k + \alpha_P d_y \in [y^k/2, 3y^k/2]$.

Finally, using the fact that $s^{k+1} \in s^k[3/4, 5/4]$ and $s^{k+1} \in y^k[3/4, 5/4]$ we have:

$$\frac{s^{k+1}y^{k+1}}{s^k y^k} \in [1/2, 3/2]$$

And since $\frac{s^k y^k}{\mu^k} \in [1/2, 3/2]$ we have $\frac{s^{k+1}y^{k+1}}{\mu^k} \in [1/4, 3]$ which concludes the proof. □

3 Practical algorithm

$$D_\gamma(x, y) = \frac{\|\nabla L(x, y)\|_\infty}{\sqrt{\|y\|_\infty + 1}}$$
$$E_\mu(x, y, s) = \max\{D(x, y), \|Sy - \mu\|_\infty\}$$

Algorithm 2 One phase primal-dual IPM

function IPM(x, y)
  $\gamma \leftarrow ...$
  $f_\gamma(x) := f(x) + \gamma\|x\|^2$
  for $k = 1, ...\infty$ **do**
    Form primal schur complement
    Factorize using IPOPT strategy.
    for $i = 1, ..., p(\#corrections)$ **do**
      if $i > 1$ & $D_\gamma(x, y) \leq \min\{\mu, \theta\}$ & $\frac{Sy}{\mu} \in [\beta_2, 1/\beta_2]$ & $\frac{\sqrt{\lambda}}{1+\|y\|_\infty} \leq \mu$ **then**
        AGGRESSIVE-CORRECTION
        $\gamma \leftarrow ...$
        If failure and $i = 2$ then set failure flag.
      else
        STABLE-CORRECTION
        If failure and $i = 1$ then set failure flag.
      end if
    end for
  end for
end function

3.1 Trust region

Algorithm 3 Stable-trust-region-step

function STABLE-TRUST-REGION-STEP(f, a, x, y, r)
  for $j = 1, ..., \infty$ **do**
    $(d_x, d_y, M^+) \leftarrow$ APPROX-PRIMAL-DUAL-TRUST-REGION(f, a, x, y, r)
    $x^+, y^+, \alpha^+ \leftarrow$ stable-line-search(...)
    if $\alpha^+ > \alpha_{\min}$ **then**
      if $j = 1$ & $\alpha = 1$ **then**
        $r^+ \leftarrow 10r$
      else if $\alpha^+ < \alpha_{small}$ **then**
        $r^+ \leftarrow r/2$
      else
        $r^+ \leftarrow \|d_x\|_2$
      end if
      break
    end if
    $r^+ \leftarrow r^+/8$
  end for
  return (x^+, y^+, M^+, r^+)
end function

Algorithm 4 Stable-trust-region-step

function STABLE-TRUST-REGION-IPOPT-STYLE(f, a, x, y, r)
  for $j = 1, ..., \infty$ **do**
    Factorize M with ... δ ...
    if inertia is good **then**
      ...
    end if
    $x^+, y^+, \alpha^+ \leftarrow$ stable-line-search(...)
  end for
  return (x^+, y^+, M^+, r^+)
end function

3.2 Line searches

function MOVE$(f, a, x, y, d_x, d_y, \eta, \alpha_P)$
  $x^+ \leftarrow x + \alpha_P d_x$
  $\mu^+ \leftarrow (1 - \eta\alpha_P)\mu$
  $\theta^+ \leftarrow (1 - \eta\alpha_P)\theta$
  $s^+ \leftarrow a(x^+) + \theta(s^1 - a(x^1))$
  $\alpha_D \leftarrow \arg\max_{\alpha \in [0,1]} \alpha$ s.t. $\frac{S^+(y+d_y\alpha_D)}{\mu} \in [e\beta_1, e/\beta_1]$
  $y^+ \leftarrow y + \alpha_D d_y$
end function

function DUAL-LINE-SEARCH$(f, a, x^+, s^+, y, d_y, \alpha_P)$
  u_D, **status**1 $\leftarrow \arg\max_{\alpha \in [0,1]} \alpha$ s.t. $\frac{S^+(y+d_y\alpha_D)}{\mu} \in [e\beta_1, e/\beta_1]$
  l_D, **status**2 $\leftarrow \arg\min_{\alpha \in [0,1]} \alpha$ s.t. $\frac{S^+(y+d_y\alpha_D)}{\mu} \in [e\beta_1, e/\beta_1]$
  $\alpha_D \leftarrow \arg\min_{\alpha \in [l_D, u_D]} D_\gamma(x^+, y + \alpha d_y)$
  $y^+ \leftarrow y + \alpha_D d_y$
end function

Algorithm 5 Stable line search

function STABLE-LINE-SEARCH$(f, a, x, y, s, d_x, d_y, d_s\eta)$
  $\eta \leftarrow 0$
  $\alpha_P \leftarrow$ FRACTIONTOBOUNDARY(y, s, d_y, d_s)
  for $i = 1, ..., \infty$ **do**
    x^+, y^+, **status** $\leftarrow$ MOVE$(f, a, x, y, d_x, d_y, \eta, \alpha_P)$
    if status = feasible then
      if sufficient progress on merit function **then**
        return x^+, y^+
      else

      end if
    else
      $\alpha_P \leftarrow \alpha_P / 2$
    end if
    ...
  end for
end function

4 Scrap paper

5 Log barrier sub-problems

This paper is concerned with the following problem:

$$\min f(x) - \mu \log(s) + \frac{1}{2} d_x^T D_x d_x + \frac{1}{2} d_s^T D_s d_s \tag{8a}$$

$$a(x) - s = r\mu \tag{8b}$$

$$s \geq 0 \tag{8c}$$

The KKT conditions for (8) are:

$$\nabla_x \mathcal{L}(x,y) = \nabla f(x) + D_x d_x - \nabla a(x)^T y = 0 \tag{9a}$$

$$\mathcal{C}_\mu(s,y) = Ys - \mu e = 0 \tag{9b}$$

$$\mathcal{P}_\mu(x,s) = a(x) - s - \mu r = 0 \tag{9c}$$

$$s, y \geq 0 \tag{9d}$$

Where the Lagrangian $\mathcal{L}(x,y) := f(x) - y^T a(x)$.

We combine the log barrier merit function and the complementary conditions as follows:

$$\phi(x,y) = \psi(x) + \zeta(x,y) \tag{10}$$

With:

$$\zeta(x,y) = \frac{\|\mathcal{C}(x,y)\|_\infty^3}{\mu^2}$$

We now introduces models to locally approximate these merit functions $\nabla_x \mathcal{L}(x,y)$, ψ, $\mathcal{C}$ and ϕ respectively. To describe our approximations of a function f around the point (x,y) we use the function $\tilde{\Delta}^f_{(x,y)}(u,v)$ to denote the predicted increase in the function f at the new point $(x+u, y+v)$. Observe that we use different approximations depending on the choice of function f.

We use a typical linear approximate of $\nabla_x \mathcal{L}(x,y)$ as follows:

$$\tilde{\Delta}^{\nabla_x \mathcal{L}}_{(x,y)}(d_x, d_y) = \nabla_{x,x} L(x,y) d_x + \nabla a(x) d_y \tag{11}$$

The following function $\tilde{\Delta}^\psi_{(x,y)}(u)$ is an approximation of the function $\psi(x)$ at the point (x,y) and predicts how much the function ψ changes as we change the current from x to $x+u$.

$$\tilde{\Delta}^\psi_{(x,y)}(u) = \frac{1}{2} u^T M(x,y) u + \nabla \psi(x)^T u \tag{12}$$

With:

$$M(x,y) = \nabla^2 \mathcal{L}(x,y) + \sum_i \frac{y_i}{a(x)} \nabla a(x)^T \nabla a(x) \tag{13}$$

Note that if we set $y_i = \frac{\mu}{s_i}$ then $M(x,y) = \nabla^2 \psi(x)$ and $\tilde{\Delta}^\psi_{(x,y)}$ becomes the second order taylor approximation of ψ at the point x. Thus we can think of $\tilde{\Delta}^\psi_{(x,y)}(u)$ as a primal-dual approximation of the function ψ.

We can also build a model of the $\zeta(x,y)$ as follows:

$$\tilde{\Delta}^\zeta_{(x,y)}(d_x, d_y) = \frac{\|Sy + Yd_s + Sd_y - \mu e\|_\infty^3 - \|\mathcal{C}(x,y)\|_\infty^3}{\mu^2} \tag{14}$$

With S a diagonal matrix containing entries of $a(x)$ and $d_s = \nabla a(x) d_x$. This model $\tilde{\Delta}^{\mathcal{C}}_{(x,y)}$ corresponds to the typical primal-dual linear model of $\mathcal{C}$ i.e. $C(x+d_x, y+d_y) \approx Sy + Yd_s + Sd_y - \mu e$.

With S and Y contain the diagonal elements of $a(x)$ and y respectively.

This allows us to approximate the change in the function ϕ at the point (x,y) as follows:

$$\tilde{\Delta}^\phi_{(x,y)}(d_x, d_y) = \tilde{\Delta}^\psi_{(x,y)}(d_x) + \tilde{\Delta}^\zeta_{(x,y)}(d_x, d_y) \tag{15}$$

We say an iterate (x, y) satisfies approximate complementary if $(x, y) \in \mathcal{Q}_\mu$ where $\mathcal{Q}_\mu$ is defined as follows:

$$\mathcal{Q}_\mu = \left\{(x, y) \in R^n \times R^m : a(x) > 0, y > 0, \|\mathcal{C}(x, y)\|_\infty \leq \frac{\mu}{2}\right\} \tag{16}$$

We say the point (x, y) is a μ-scaled KKT point if $(x, y) \in \mathcal{T}_\mu$ where:

$$\mathcal{T}_\mu = \{(x, y) \in \mathcal{Q}_\mu : \|\nabla \mathcal{L}(x, y)\| \leq \mu(\|y\|_1 + 1)\} \tag{17}$$

In which case the algorithm terminates.

6 Algorithm

Let S, Y denote the diagonal matrices with entries of s and y respectively. We can linearize (9) at the iterate (x, y, s) as follows:

$$\begin{bmatrix} \nabla^2 \mathcal{L}(\hat{x}, \hat{y}) + D_x & -\nabla a(\hat{x})^T & 0 \\ \nabla a(\hat{x}) & 0 & -I \\ 0 & \hat{S} & \hat{Y} + D_s \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_s \end{bmatrix} = - \begin{bmatrix} \nabla \mathcal{L}(x, y) \\ \mathcal{P}_\mu(x, s) \\ \mathcal{C}_\mu(s, y) \end{bmatrix} \tag{18}$$

Which is equivalent to solving:

$$\begin{bmatrix} \nabla^2 \mathcal{L}(\hat{x}, \hat{y}) + \nabla a(x)^T D_s \nabla a(x) + D_x & \nabla a(\hat{x})^T \\ \nabla a(\hat{x}) & -(\hat{Y} + D_s)^{-1} \hat{S} \end{bmatrix} \begin{bmatrix} d_x \\ -d_y \end{bmatrix} = - \begin{bmatrix} \nabla \mathcal{L}(x, y) \\ \mathcal{P}_\mu(x, s) + (\hat{Y} + D_s)^{-1} \mathcal{C}_\mu(s, y) \end{bmatrix} \tag{19}$$

One can also solve this system by solving the Schur complement:

$$(\nabla^2 \mathcal{L}(\hat{x}, \hat{y}) + \nabla a(\hat{x})^T (\hat{Y} + D_s) \hat{S}^{-1} \nabla a(\hat{x}) + D_x) d_x = -\nabla \mathcal{L}(x, \mu S^{-1} e) - \nabla a(\hat{x})^T \hat{Y} \hat{S}^{-1} \mathcal{P}_\mu(x, s)$$

Observe that (**??**) may be singular or correspond to a direction that makes the log barrier objective worse. To rectify this problem we compute the direction as follows:

$$d_x = \arg \min_{\|u\|_2 \leq r} \tilde{\Delta}^\psi_{(x,y)}(u) \tag{20a}$$

$$d_s = \nabla a(x) d_x \tag{20b}$$

$$d_y = -S^{-1} \left(Y d_s + \mathcal{C}(x, y)\right) \tag{20c}$$

******CAREFUL WITH SIGNS i.e. should be $d_s = -\nabla a(x) d_x$, $d_y = -S^{-1} \left(Y d_s + \mathcal{C}(x, y)\right)$ ***************

It is well-known from trust region literature that there exists some $\delta \in [0, \infty)$ such that:

$$(M(x, y) + \delta I) d_x = -\nabla \psi(x) \tag{21}$$

Furthermore, by re-arranging this equation we can deduce that (d_x, d_y, d_s) satisfies a perturbed version of (**??**):

$$\begin{bmatrix} \nabla^2 \mathcal{L}(x, y) + \delta I & -\nabla a(x)^T & 0 \\ -\nabla a(x) & 0 & I \\ 0 & S & Y \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_s \end{bmatrix} = - \begin{bmatrix} \nabla \mathcal{L}(x, y) \\ 0 \\ \mathcal{C}(x, y) \end{bmatrix} \tag{22}$$

Algorithm 6 Primal-dual trust region step

function PRIMAL-DUAL-TRUST-REGION(x, y, r) **** $\in$ ****

$$d_x \in \arg \min_{\|u\| \leq r} \tilde{\Delta}^\psi_{(x,y)}(u) \tag{23a}$$

$$d_s = \nabla a(x) d_x \tag{23b}$$

$$S = \text{Diag}(a(x)) \tag{23c}$$

$$d_y = -S^{-1} \left(Y d_s + \mathcal{C}(x, y)\right) \tag{23d}$$

$(x^+, y^+) \leftarrow (x + d_x, y + d_y)$

return(x^+, y^+, d_x, d_y)

end function

Our complete algorithm is summarized as follows:

Algorithm 7 Primal-dual non-convex interior point algorithm

function NON-CONVEX-IPM(x^1, y^1)
  for $k = 1, ..., \infty$ **do**
    $r \leftarrow R(y^k)$
    repeat
      $(x^+, y^+, d_x, d_y) \leftarrow$ PRIMAL-DUAL-TRUST-REGION(x^k, y^k, r)
      if $(x^+, y^+) \in \mathcal{Q}_\mu$ **then**
        if $(x^+, y^+) \in \mathcal{T}_\mu$ **then**
          return (x^+, y^+)
        end if
      end if
      $r \leftarrow r/2$
    until $\phi(x^+) > \phi(x^k) + \frac{1}{2}\tilde{\Delta}^{\phi}_{(x^k,y^k)}(d_x, d_y)$
    $x^k \leftarrow x^+$
    $y^k \leftarrow y^+$
  end for
end function

7 Delta computation

Algorithm 8 Delta

$\gamma_{lb} = 0$, $\gamma_{ub} = \delta_{\max} = \|H\|_F^2$, δ_{k-1} ▷ lower and upper bounds on minimum eigenvalue
Try $\delta = 0$, if succeeds, trial solve with this delta. If step size is small skip to trust region step.
$\delta = \delta_{k-1}$
if $\delta = 0$ **then**
  $\delta = \delta_{\min}$
end if
for $i = 1, ..., \infty$ **do**
  Break if inertia correct and update γ_{lb} and γ_{ub}.
  $\delta = \delta 100$
end for
Trust region
$R = \|d_x^{k-1}\|_2$
for $i = 1, ..., \infty$ **do**
  Compute trust region with R
  If trust region is too accurate increase radius size
  If step unsuccessful decrease radius size
  Prevent oscillation
end for