A one phase IPM for non-convex optimization

Oliver Hinder, Yinyu Ye

September 6, 2017

Abstract

The work of [Wächter and Biegler, 2000] suggests that infeasible start interior point methods (IPMs) developed for linear programming cannot be adapted to non-linear optimization without significant modification i.e. using a two phase or a penalty method. We propose an IPM, that by careful initialization and updates of the slack variables, is guaranteed to find an first order certificate of local infeasibility, local optimal or unboundedness. Our proposed algorithm differs from other IPM methods for non-convex programming, because we reduce primal feasibility at the same rate as the barrier parameter. This gives an algorithm with more robust convergence properties and closely resembles successful algorithms from linear programming. We implement the algorithm and compare with IPOPT subset of CUTEst problems. Our algorithm requires has a similar median number of iterations, however, fails on only 9% compared with 16% for IPOPT. Experiments on infeasible variants of CUTEst indicate superior performance for detecting infeasibility.

1 Introduction

This paper develops an interior point method for finding stationary points of the problem:

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

$$a(x) \leq 0, \tag{2}$$

where the functions $a(x) : \mathbb{R}^n \to \mathbb{R}^m$ and $f : \mathbb{R}^n \to \mathbb{R}$ are twice differentiable and might be non-convex. Examples of real world problems in this framework include truss design, robot control, aircraft control and aircraft design [Gould et al., 2015a, TRO11X3, ROBOT, AIRCRAFTA, AVION2].

Interior point methods were first developed for linear programming [Karmarkar, 1984]. The idea for primal-dual interior point methods originates with [Megiddo, 1989]. Initially, algorithms that required a feasible starting point were studied [Kojima et al., 1989, Monteiro and Adler, 1989]. However, generally one is not given an initial point that is feasible. A naive solution to this issue is to move the constraints into the objective by adding a large penalty for constraint violation (Big-M method) [McShane et al., 1989]. A more effective solution is the infeasible start algorithm of [Lustig, 1990] which has less numerical issues and a smaller iteration count than the big-M method of [McShane et al., 1989]. This approach also simplified the algorithm, by avoiding the need to make a good initial guess for the size of the penalty parameters. Lustig's approach was further improved in the predictor-corrector algorithm of [Mehrotra, 1992]. This algorithm reduced complementarity, duality and primal feasibility at the same rate, using an adaptive heuristic. This class of methods was shown by [Todd, 2003] to converge to either optimality or infeasibility certificates (of the primal or dual).

This infeasible start approach of [Lustig, 1990] for linear programming naturally to non-linear optimization. And most interior point codes for non-convex optimization are built off these ideas [Vanderbei, 1999, Wächter and Biegler, 2006, Byrd et al., 2006]. However, [Wächter and Biegler, 2000] showed for the problem

$$\min x \tag{3a}$$

$$x^2 - s_1 - 1 = 0 \tag{3b}$$

$$x - s_2 - 1/2 = 0 \tag{3c}$$

$$s_1, s_2 \geq 0, \tag{3d}$$

a large class of infeasible start algorithms fails to converge to either a local optimum or infeasibility certificate starting at any point with $x < 0$, $s_1, s_2 > 0$. Following this paper, a flurry of research was published suggesting different methods for resolving this issue [Benson et al., 2004]. The main two approaches can be split into penalty methods [Liu and Sun, 2004, Chen and Goldfarb, 2006, Curtis, 2012, Gould et al., 2015b] and two phase algorithms [Wächter and Biegler, 2006].

Penalty methods move some measure of constraint violation into the objective. These methods require a penalty parameter M that measures how much the constraint violation contributes to the objective. For large enough M the algorithm will converge to an optimal solution. However, estimating this penalty parameter is difficult – too small and the algorithm will not find a feasible solution, too big and the algorithm might be slow and suffer from numerical issues. Consequently, typically penalty methods tend to be slow [Curtis, 2012, Algorithm 1] or use complex schemes for dynamically updating the penalty parameter [Curtis, 2012, Algorithm 2].

The algorithm IPOPT is an example of a two phase algorithm: it has a main phase and a feasibility restoration phase [Wächter and Biegler, 2006]. The main phase searches simultaneously for optimality and feasibility using a classical infeasible start method. The other phase, known as the feasibility restoration phase, aims to minimizes infeasibility. The feasibility restoration phase is only called when the main phase fails e.g. the step size is small. It is well-known that this approach has drawbacks. The algorithm has difficulties detecting infeasibility [Huang and Mehrotra, 2016, Table 15] and will fail if the feasibility restoration phase is called too close to the optimal solution. Some of these issues have been addressed by [Nocedal et al., 2014].

The main contribution of this paper is an infeasible start method interior point method for non-linear programming that builds on the work of [Lustig, 1990, Mehrotra, 1992] for linear programming. The algorithm avoids a big-M or a two phase approach. Furthermore, our solution to the issue in example (3) of [Wächter and Biegler, 2000] is simple: we carefully initialize the slack variables and use non-linear updates to ensure we approach feasibility from above. Consequently, under general conditions we guarantee that our algorithm will converge to either a local certificate of optimality, local infeasibility or unboundedness. Our algorithm has other desirable properties. Complementarity moves at the same rate as primal feasibility. This implies from [Gabriel Haeser, 2017] that, if certain sufficient conditions for local optimality conditions hold, our approach guarantees dual multipliers sequence will remain bounded. In contrast, methods that reduce the primal feasibility too quickly, such as IPOPT, the dual multiplier sequence can be unbounded even for linear programs.

Our method has further similarities with Mehrotra's [Mehrotra, 1992] predictor-corrector algorithm for linear programming: the rate that we reduce the dual feasibility, primal feasibility and complementarity is adaptive. We implement the algorithm and compare our solver with IPOPT on large scale CUTEst problems. Our algorithm requires has a similar median number of iterations, however, seems to fail less often. Experiments on infeasible variants of CUTEst indicate superior performance for detecting infeasibility.

The paper is structured as follows: Section 2 describes the algorithm, Section 3 gives the convergence proofs, Section 4 delves into implementation details and Section 5 presents test results on the CUTEst test set.

A. talk about sequence stuff or MFCQ [MOVE??]

B. talk about filter methods and how people observed better performance by avoiding penalty functions (so they take bigger steps)

Summary of contributions:

A. Adaptive heuristics for reduction in μ

2 The algorithm

Consider a naive log barrier problems of the form:

$$\min_{x \in \mathbb{R}^n} f(x) - \mu \sum_i \log\left(-a_i(x)\right) \tag{4}$$

The idea is to solve sequence of these sub-problems with $\mu \to 0$ and $\mu \geq 0$. The log barrier transforms the non-differentiable original problem (1) into a twice differentiable log barrier function on which we can apply newton's method. However, there are issues with this naive log barrier formulation. We are rarely given a feasible starting point. Furthermore, one would like to ensure that the primal and dual variables remain bounded. To resolve these issues we consider the shifted and regularized sub-problem described as follows:

$$\min_{x \in \mathbb{R}^n} \psi_\mu(x) := f(x) + \mu r(x) - \mu \sum_i \log\left(\mu w_i - a_i(x)\right) \tag{5}$$

With some vector $w \geq 0$ which remains fixed for all sub-problems, and some $\mu > 0$ which measures the size of the shift. The function $r : \mathbb{R}^n \to \mathbb{R}$ is define by:

$$r(x) := \beta_{10} \sum_{i=1}^{n} \sqrt{x_i^2 + 1/\beta_{10}^2} - \beta_{11} e^T a(x). \tag{6}$$

where $\beta_{10}, \beta_{11} \in (0, 1)$ are constants. The purpose of r is to prevent the primal iterates from unnecessarily diverging.

Holistically, our technique consists of computing two types of directions: stabilization and aggressive directions. Both of these directions are computed from the same linear system with different right hand sides. Aggressive directions are equivalent to affine scaling steps [Mehrotra, 1992]; since they apply a newton step directly to the KKT system, ignoring the barrier parameter μ. Aggressive steps aim to simultaneously approach optimality and feasibility. However, continuously taking aggressive steps may cause the algorithm to stall or fail to converge. To remedy this we have a stabilization step. The stabilization steps keeps the primal feasibility the same i.e. uses $\eta = 0$ and aims to reduce the log barrier objective until an approximate solution to the shifted log barrier problem is found. While this step has similar goals to the centering step of Mehrotra there are distinct differences. The centering steps of Mehrotra move the iterates towards the central path, while keeping the primal and dual feasibility fixed. However, our stabilization steps only keep the primal feasibility fixed, while reducing the log barrier objective. This choice reflects challenges that occur in non-convex optimization and will be discussed more in [??]

The interior point method that we develop generates a sequences of primal iterates x^k, $s^k \in \mathbb{R}^n$ with $s^k > 0$, barrier parameter $\mu^k > 0$ and feasibility violation $\mu^k > 0$ that satisfy:

$$a(x^k) + s^k = \mu^k w \tag{7a}$$

$$\frac{S^k y^k}{\mu^k} \in [e\beta_1, e/\beta_1], \tag{7b}$$

Where $w \geq 0$ is vector selected such that at the initial point $a(x^0) + s^0 = \mu^0 w$ and $\beta_1 \in (0, 1)$ is an algorithmic parameter. This set of equations implies the primal feasibility and complementarity are moved at the same rate. Furthermore, there exists a subsequence of the iterates (those that satisfy the aggressive step criterion (24)) such that:

$$\frac{\|\nabla_x \mathcal{L}(x^k, y^k)\|_\infty}{\mu^k(\|y^k\|_\infty + 1)} \leq 1, \tag{8}$$

where $\mathcal{L}(x, y) := f(x) + y^T a(x)$ is the Lagragian function. Equations (7) and (8) holding is common in many practical linear programming implementations [Mehrotra, 1992, **?**]. These conditions holding is desirable, because it implies the dual variables likely remained bounded. To be more precise, assume the subsequence satisfying (7) and (8) is converging to a feasible solution. If this solution satisfies certain sufficiency conditions for local optimality, then as shown in [Gabriel Haeser, 2017] the dual variables will remain bounded. Note that (7) and (8) can be interpreted as a 'central sequence'. This is weaker than the existence of a central path, a concept from convex optimization [Megiddo, 1989, Andersen and Ye, 1999]. Unfortunately, in non-convex optimization there may not exist a continuous central path (see Appendix B).

interpretation of equality v.s. inequality constraints primal-dual methods

2.1 Derivation of direction computation

We cannot apply a newton method directly to problem (5) without adding a proximal term (9a) parameterized by δ and centered about the previous iterate x. This ensures the newton direction exists and is decreasing in the objective value for a fixed constraint violation i.e. $\eta = 0$. Therefore, we find the next iterate by approximately solving:

$$x^*, s^* \leftarrow \arg \min_{\bar{x}\in\mathbb{R}^n, \bar{s}\in\mathbb{R}^{m++}} f(\bar{x}) - (1-\eta)\mu \left(-r(\bar{x}) + \sum_i \log \bar{s}_i \right) + \frac{\delta}{2}\|\bar{x} - x\|^2 \tag{9a}$$

$$a(\bar{x}) + \bar{s} = (1-\eta)\mu w \tag{9b}$$

$$\bar{s} \geq 0. \tag{9c}$$

Writing the first-order local optimality conditions for this problem gives:

$$\begin{aligned} \nabla_x \mathcal{L}(x^*, y^*) + \delta(x^* - x) &= (1-\eta)\mu \nabla r(x^*) \\ a(x^*) + s^* &= (1-\eta)\mu w \\ s_i^* y_i^* &= (1-\eta)\mu \\ s^*, y^* &\geq 0 \end{aligned}$$

Primal interior point methods [Fiacco and McCormick, 1990] apply Newton's method directly to system (9). However, they have inferior practical performance to primal-dual methods, that apply newton's method directly to the optimality conditions. Therefore, we use primal-dual search directions defined as follows:

$$\mathcal{K}_\delta d = -b \tag{10}$$

where

$$d = \begin{bmatrix} d_x \\ d_s \\ d_y \end{bmatrix} \tag{11}$$

$$b = \begin{bmatrix} \nabla_x \mathcal{L}(x,y) + (1-\eta)\mu\nabla r(x) \\ \eta\mu w \\ Ys - \eta\mu e \end{bmatrix} \tag{12}$$

$$\mathcal{K}_\delta = \begin{bmatrix} \nabla_x^2 \mathcal{L}(x,y) + (1-\eta)\mu\nabla^2 r(x) + \delta I & \nabla a(x)^T & 0 \\ \nabla a(x) & 0 & I \\ 0 & S & Y \end{bmatrix}. \tag{13}$$

We use $\mathcal{K}_\delta^{-1}$ to denote the factorization of $\mathcal{K}_\delta$ e.g. LDL.
By taking the primal schur complement one can see solving system (10) for d_x is equivalent to solving:

$$(\mathcal{M} + \delta I)d_x = -\nabla f(x) - \nabla a(x)^T((1-\eta)\mu S^{-1}e + \eta Y(e + \mu S^{-1}w) - \mu(1-\eta)\nabla r(x) \tag{14}$$

Where the matrix $\mathcal{M}$ is

$$\mathcal{M} = \nabla_x^2 \mathcal{L}(x,y) + (1-\eta)\mu\nabla^2 r(x) + \nabla a(x)^T Y S^{-1} \nabla a(x), \tag{15}$$

is a primal-dual approximation of the hessian of the log barrier function $\psi_\mu(x)$. Note that if the matrix $(\mathcal{M} + \delta I)$ is positive definite and $\eta = 0$ then the right hand side of (14) becomes $-\nabla\psi_\mu(x)$ hence the direction d_x is a descent direction on the function $\psi_\mu(x)$. Consequently, during our algorithm we will pick $\delta > 0$ such that the matrix $(\mathcal{M} + \delta I)$ is positive definite.

careful with $\nabla^2 r(x)$.

2.2 Updating the iterates

Suppose that we have computed some direction d by solving system (10). We wish to construct a candidate (x^+, s^+, y^+, μ^+) for the next iterate. Given a primal step size α_P we update the primal iterates as follows:

$$\mu^+ \leftarrow (1 - \eta\alpha_P)\mu \tag{16a}$$

$$x^+ \leftarrow x + \alpha_P d_x \tag{16b}$$

$$s^+ \leftarrow \mu^+ w - a(x^+) \tag{16c}$$

The slack variable update (16c) is non-linear and its purpose is to ensure that equation (7a) remains satisfied and therefore we can control the rate of reduction of primal feasibility. However, if the function a is linear the slack variable update (16c) reduces to:

$$s^+ = \mu^+ w - a(x) - \alpha_P \nabla a(x) d_x = (\mu w - a(x)) - \alpha_P \left(\eta\mu w + \nabla a(x) d_x\right) = s + \alpha_P d_s$$

Therefore if the function a is linear we use the same updates as infeasible start algorithms for linear programming [Lustig, 1990, Mehrotra, 1992]. We remark that non-linear updates for the slack variables have been used in other interior point methods [Curtis, 2012].

Next, we specify a criterion to prevent the slack variables from getting too close to the boundary. In particular, given any candidate primal iterate (x^+, s^+) we require that the following fraction to the boundary rule is satisfied:

$$s^+ \geq \beta_7 \min\{s, \|d_x\|_\infty^2\} \tag{17}$$

Finally, it remains to desribe how to update the dual variables. Given some candidate primal iterate x^+, s^+, then let $B(s^+, d_y)$ be the set of feasible dual step sizes. More precisely, $B(s^+, d_y)$ is the largest interval such that if $\alpha_D \in B(s^+, d_y)$ then

$$\frac{S^+(y + \alpha_D d_y)}{\mu^+} \in [\beta_1, 1/\beta_1] \tag{18a}$$

If no such interval exists we set $B(s^+, d_y)$ to the empty set and the step will be rejected. We compute the dual step size as follows:

$$\alpha_D = \arg \min_{\alpha_D \in B(s^+, d_y)} \|S^+ y - \mu^+ + \alpha_D S^+ d_y\|_2^2 + \|\nabla_x \mathcal{L}(x, y) + (\nabla_x^2 \mathcal{L}(x, y) + \delta I)\alpha_P d_x + \alpha_D \nabla a(x)^T d_y\|_2^2. \tag{19}$$

This reduces to a one dimensional least squares problem in α_D which has a closed form expression.

2.3 Termination criterion

Now, we have derived the primal infeasibility termination criterion we can present the termination criterion for our algorithm. Define the following function σ,

$$\sigma(y) := \frac{100}{\max\{100, \|y\|_\infty\}}$$

which is a scaling factor based on the size of the dual variables. This scaling factor is related to s_d and s_c in the IPOPT implementation paper [Wächter and Biegler, 2006]. We use $\sigma(y)$ in the local optimality termination criterion (20) because there may be numerical issues reducing the unscaled dual feasibility if the dual multipliers become large. In particular, the first order optimality termination criterion we use is:

$$\sigma(y)\|\nabla \mathcal{L}(x, y)\|_\infty \leq \epsilon_{\mathbf{opt}} \tag{20a}$$

$$\sigma(y)\|Sy\|_\infty \leq \epsilon_{\mathbf{opt}} \tag{20b}$$

$$\|a(x) + s\|_\infty \leq \epsilon_{\mathbf{opt}}. \tag{20c}$$

The first order local primal infeasibility termination criterion is given by:

$$\Gamma(x, y, s, \mu) \leq \epsilon_{\mathbf{inf}} \tag{21}$$

where

$$\Gamma(x, y, s, \mu) := \frac{\max\{\|\nabla a(x)^T y\|_\infty, \|Sy\|_\infty\}}{\|Yw\|_\infty \min\{1, \mu\}}. \tag{22}$$

We remark that if we find a point with $\Gamma(x, y, s, \mu) = 0$ then we if $w = e$ have found a stationary point to the problem:

$$\min_{x \in \mathbb{R}^n} \max_{i \in \{1,\ldots,m\}} a_i(x)$$

For a more thorough justification of this choice for the infeasibility termination criterion see Section 3.1.

The unboundedness termination criterion is given by

$$\frac{\max\{\|a(x)^+\|_\infty, 1\}}{\min\{\max\{1, -f(x)\}\}, \|x\|_\infty\}} \leq \epsilon_{\mathbf{unbd}}. \tag{23}$$

Satisfying this termination criterion for arbitrary small $\epsilon_{\mathbf{unbd}}$ does not guarantee that the problem has an objective that is unbounded from below on the feasible region. However, if the functions f and a are convex, and there exists a strictly feasible solution, then if the criterion is satisfied as $\epsilon_{\mathbf{unbd}} \to 0$ one can conclude the objective is unbounded from below on the feasible region.

2.4 Algorithm outline

Before we outline our algorithm, we need to define the switching condition when take an aggressive direction instead of a stabilization step, which we define as follows:

$$\sigma(y)\|\nabla \mathcal{L}(x, y)\|_\infty \leq \mu \tag{24a}$$

$$\|\nabla \mathcal{L}(x, y)\|_\infty \leq \|\nabla f(x)\|_\infty + \mu/\beta_2 \tag{24b}$$

$$\frac{Sy}{\mu} \in [e\beta_2, e/\beta_2]. \tag{24c}$$

where the parameter $\beta_2 \in (0, \beta_1)$. The purpose of (24a) is to ensure that we have approximately solved the shifted log barrier problem. Equation (24b) also helps ensure (as we show in Section 3.3) that if the dual variables are diverging rapidly then the infeasibility termination criterion is met. Finally, equation (24c) with $\beta_2 < \beta_1$ ensures we have some a buffer in the complementarity such that we can still satisfy (18a) when we move in the aggressive search direction. Algorithm 1 formally outlines our one phase interior point method. Note that Algorithm 1 does not include the details for the aggressive or stabilization steps that are given in Algorithm 2 and Algorithm 3 respectively. Since Algorithm 1 maintains $a(x) + s = w\mu$ for each iterate, it requires the starting point satisfies:

$$a(x^0) + s^0 = w\mu^0,$$

with $w \geq 0$ and $\mu^0 > 0$. For any fixed x^0 one can always pick sufficiently large w and μ^0 such that $\mu^0 w > a(x^0)$ and setting $s^0 \leftarrow w\mu^0 - a(x^0)$ meets our requirements. For the details of how initialize the variables in the practical implementation see Section 4.1.

The general idea of Algorithm 1 is as follows. At each outer iteration we factorize the matrix $\mathcal{K}_\delta$ with an appropriate choice of δ using Algorithm 5 (based off ideas of IPOPT). With this factorization fixed, we then attempt to take multiple inner iterations (at most $c_{\max}$), which corresponds solving system (10) with different right hand sides choices, but the same matrix $\mathcal{K}_\delta$. Each inner iterations is either an aggressive or stabilization steps. If, on the inner iteration, the step fails (i.e. due to a too small step size), then we increase δ to address this failure and re-factorization $\mathcal{K}_\delta$. Note that we evaluate the Hessian of the Lagrangian and the full Jacobian once per outer iteration (we do one Jacobian-vector product per inner iteration).

Algorithm 1 High level description of one phase IPM

Input: some initial point x^0, vector $w \geq 0$ and variables $y^0, s^0, \mu^0 > 0$ such that $a(x^0) + s^0 = w\mu^0$. Termination tolerances $\epsilon_{\mathbf{inf}}$, $\epsilon_{\mathbf{unbd}}, \epsilon_{\mathbf{inf}} > 0$.
Output: some point (x, y, s, μ) that satisfies the termination criterion (inequalities (20), (21) or (23))

For each outer iteration $i \in \{1, \ldots, i_{\max}\}$ perform the following steps:

A.1 *Evaluate the matrix Hessian of the Lagrangian $\nabla_x^2 \mathcal{L}(x, y)$ and the Jacobian of the constraints $\nabla a(x)$.*

A.2 *Form the matrices $\mathcal{K}_0$ and $\mathcal{M}$ using* (13) *and* (15) *respectively, at the point (x, y, s, μ) with $\eta = 1$ if the aggressive step criterion* (24) *is satisfied and $\eta = 0$ otherwise.*

A.3 *Select δ and factorize the matrix $\mathcal{K}_\delta$,*
i.e. run Algorithm 5 with:
Input: $\mathcal{K}_0$, δ.
Output: New value for δ, factorization $\mathcal{K}_\delta^{-1}$.

A.4 *Perform inner iterations.* For $j \in \{1, \ldots, c_{\max}\}$:

A.4.1 *Take step*

-Case-I If the aggressive step criterion (24) is satisfied, do an aggressive step,
i.e. run Algorithm 2 with:
Input: $\mathcal{K}_\delta, \mathcal{K}_\delta^{-1}$ and the point (x, y, s, μ).
Output: A **status** and a new point (x^+, y^+, s^+, μ^+).

-Case-II Otherwise, do a stabilization step,
i.e. run Algorithm 3 with:
Input: $\mathcal{K}_\delta, \mathcal{K}_\delta^{-1}, \mathcal{M}$ and the point (x, y, s, μ).
Output: A **status** and a new point (x^+, y^+, s^+, μ^+).

A.4.2 *Deal with failures.* If **status** = SUCCESS set $(x, y, s, \mu) \leftarrow (x^+, y^+, s^+, \mu^+)$. If **status** = FAILURE and $j = 1$ go to (A.6). If **status** = FAILURE and $j > 1$ go to go to step (A.1).

A.4.3 *Check termination criterion.* If any of the inequalities (20), (21) or (23) hold at the point (x, y, s, μ) terminate the algorithm.

A.5 Go to (A.1).

A.6 *Increase delta to address failure.* Set $\delta = \max\{\delta_{\text{inc}}\delta, \delta_{\min}\}$ and factorize the matrix $\mathcal{K}_\delta$. Go to step (A.4).

In both the aggressive steps and stabilization steps we use a backtracking line search. We choose the initial trial primal step size $\alpha_P^{\max}$ to be the maximum $\alpha_P \in [0, 1]$ that satisfies the following fraction to the boundary rule:

$$s + \alpha_P d_s \geq \beta_8 \min\{s, \max\{\|d_x\|_\infty^2, \|d_x\|_\infty^{\beta_9}\}\} \tag{25}$$

where the parameter $\beta_8 \in (\beta_7, 1)$ and $\beta_9 \in (1, 2)$. The idea of this choice for $\alpha_P^{\max}$ is that the fraction to the boundary rule (17) is likely to be satisfied for the first trial point i.e. $\alpha_P = \alpha_P^{\max}$. This is because:

$$\beta_8 \min\{s, \max\{\|d_x\|_\infty^2, \|d_x\|_\infty^{\beta_9}\}\} > \beta_7 \min\{s, \|d_x\|_\infty^2\}$$

corresponding to the right hand side of equation (25) and equation (17) respectively.

2.5 Aggressive steps

Recall that when computing aggressive search directions we solve the system (10) with $\eta = 1$, that is, we aim for both feasibility and optimality simultaneously. We accept any step size assuming it satisfies the

fraction to the boundary rule (17) and the set of valid dual step sizes are non-empty $B(s^+, d_y) \neq \emptyset$ (see equations (18)).

The backtracking line search of the aggressive step has a minimum step size. If during the backtracking line search (line A.4 of Algorithm 2) the step size α_P satisfies:

$$\alpha_P \leq \min_{i \in \{1,\dots,m\}: w_i > 0} \frac{\beta_6 s_i}{4\mu w_i} \tag{26}$$

then we immediately reject the step and exit Algorithm 2. Following this, δ is increased in Line A.6 of Algorithm 1 and a new aggressive step is attempted. It is possible that δ will be increased many times, however, for sufficiently large δ an acceptable step will be found (see Lemma 4).

Algorithm 2 High level description of aggressive step

Input: The point (x, y, s, μ), the matrix $\mathcal{K}_\delta$ and its factorization $\mathcal{K}_\delta^{-1}$.
Output: A new point (x^+, y^+, s^+, μ^+) and a **status** of either SUCCESS or FAILURE

A.1 Compute the vector b at the point (x, y, s, μ) via (12) with $\eta = 1$.

A.2 Solve the system $\mathcal{K}_\delta d = -b$.

A.3 Estimate the largest primal step size $\alpha_P^{\max}$ from equation (25).

A.4 *Perform a backtracking line search on the primal step* α_P. Trial step sizes $\alpha_P \in \{\alpha_P^{\max}, \beta_6\alpha_P^{\max}, \beta_6^2\alpha_P^{\max}, ...\}$ computing the trial point (x^+, y^+, s^+, μ^+) as described in (16). Terminate with **status** = SUCCESS and return the trial point the first time all of the following conditions hold:

(i) The fraction to the boundary rule (17) is satisfied

(ii) The set of valid dual step sizes is non-empty i.e. $B(s^+, d_y) \neq \emptyset$

Terminate with **status** = FAILURE if the line search step size with (26) satisfied.

2.6 Stabilization step

2.6.1 Augmented log barrier merit function

When the stabilization step is called the goal is to minimize the function ψ_μ keeping the constraint violation and barrier parameter fixed, until criterion (24) for an aggressive step is met. For this reason, it makes sense to use the log barrier function ψ_μ to measure progress. We can approximate the change in the log barrier function by the following model:

$$\tilde{\Delta}_{(x,y)}^{\psi_\mu}(u) = \frac{1}{2}u^T\mathcal{M}u + \nabla\psi_\mu(x)^T u \tag{27}$$

where $\mathcal{M}$ is computed on line A.2 of Algorithm 1.

Note that if $\mathcal{M} = \nabla_\mu^2\psi(x)$ then $\tilde{\Delta}_{(x,y)}^{\psi}$ becomes the second order taylor approximation of ψ_μ at the point x. Thus, we can think of $\tilde{\Delta}_{(x,y)}^{\psi}(u)$ as a primal-dual approximation of the barrier function ψ_μ. Note that the log barrier function does not measure anything with respect to the dual iterates. This might impede performance if $\|Sy - \mu\|_\infty$ is large, but $\|\nabla\psi_\mu(x)\|$ is small. In this case, taking a large step might reduce the complementarity significantly, even though the barrier function increases slightly. Therefore we add a complementarity measure to the barrier function to create an augmented log barrier function:

$$\phi_\mu(x, y, s) := \psi_\mu(x) + \zeta(s, y), \tag{28}$$

where

$$\zeta(s, y) = \frac{\|Sy - \mu\|_\infty^3}{\mu^2}.$$

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

$$\tilde{\Delta}^{\zeta}_{(x,y)}(u,v) = \frac{\|Sy + Y\nabla a(x)u + Sv - \mu e\|^3_\infty - \|Sy - \mu\|^3_\infty}{\mu^2}$$

and our model of ϕ_μ is

$$\tilde{\Delta}^{\phi_\mu}_{(x,y)}(u,v) = \tilde{\Delta}^{\psi_\mu}_{(x,y)}(u) + \tilde{\Delta}^{\zeta}_{(x,y)}(u,v). \tag{29}$$

We say that the candidate iterates x^+, y^+, s^+ have made sufficient progress on ϕ over the current iterate x, y, s if:

$$\phi_\mu(x^+, y^+, s^+) \leq \phi_\mu(x,y,s) + \beta_5 \tilde{\Delta}^{\phi_\mu}_{(x,y)}(\alpha_P d_x, \alpha_P d_y) \tag{30}$$

where $\beta_5 \in (0,1)$ is a user defined parameter.

2.6.2 KKT merit function and filter

In the stabilization search directions we accept steps that make progress one on of two merit functions, which form a filter. The first function ϕ_μ is defined in Section 2.6.1. The second function, we call the KKT merit function, measures the scaled dual feasibility and complementary:

$$\mathbb{K}_\mu(x,y,s) = \sigma(y)\max\{\|\nabla\mathcal{L}(x,y)\|_\infty, \|Sy - \mu\|_\infty\} \tag{31}$$

This merit function measures progress effectively in regimes where the matrix $\mathcal{M}$ is positive definite. In this case, the search directions generated by (14) will be a descent direction on this merit function[1]. This merit function is similar to the types of the potential functions used in interior point methods for convex optimization [Andersen and Ye, 1998, Huang and Mehrotra, 2016]. Unfortunately, while this merit function may be an excellent choice for convex problems, in non-convex optimization it has serious issues. In particular, the search direction (14) will not be a descent direction on this merit function. Moreover, changing the search direction to minimize the dual feasibility has negative ramifications. The algorithm could to converge to a critical point of the dual feasibility where $\mathbb{K}_\mu(x,y,s) \neq 0$[2]. For further discussion of these issues see [Shanno and Vanderbei, 2000].

While it is sufficient to guarantee convergence by accepting steps if (30) is satisfied, in some regimes e.g. when $\mathcal{M}$ is positive definite, this may select step sizes α_P that are too conservative. For example, this naturally occurs near points satisfying the sufficient conditions for local optimality. In these situation often the KKT error is a better measure of progress towards a local optimum than a merit function that discards information about the dual feasibility. Furthermore, from our experience, when converging towards an optimal solution numerical errors in the function ϕ_μ may cause the algorithm to fail to make sufficient progress on the merit function ϕ_μ i.e. (30) is not satisfied for any α_P. For these reasons we decide to use a filter approach [Fletcher and Leyffer, 2002, Wächter and Biegler, 2006]. Typical filter methods [Fletcher and Leyffer, 2002] require progress on either the constraint violation or objective function. Our approach is distinctly different, because we accept steps that make progress on either the merit function ϕ_μ or the merit function $\mathbb{K}_\mu$. To be precise we accept any iterate (x^+, y^+, s^+, μ^+) that makes sufficient progress on the augment log barrier function ϕ_μ, or satisfies the following two equations

$$\mathbb{K}_\mu(x^+, y^+, s^+) \leq (1 - \beta_4\alpha_P)\mathbb{K}_\mu(\hat{x}, \hat{y}, \hat{s}) \tag{32a}$$

$$\phi_\mu(x^+, y^+, s^+) \leq \phi_\mu(\hat{x}, \hat{y}, \hat{s}) + \sqrt{\mathbb{K}_\mu(\hat{x}, \hat{y}, \hat{s})} \tag{32b}$$

for every previous iterate $(\hat{x}, \hat{y}, \hat{s}, \hat{\mu})$ with $a(\hat{x}) + \hat{s} = a(x) + s$.

The idea of (32) is that for points with similar values on the augmented log barrier function the KKT error is a good measure of progress. However, we want to discourage the algorithm from significantly increasing the augmented log barrier function while reducing the KKT error. Since, if this is occurring the algorithm might converge to a saddle point.

[1] For inner iteration $j = 1$

[2] To see why this occurs one need only consider an unconstrained problem e.g. minimizing the function $f(x) = x^4 + x^3 + x$ subject to no constraints. The point $x = 0$ is a stationary point for the gradient of $\nabla f(x)$, but is not a critical point of the function.

2.6.3 Stabilization step algorithm

During the backtracking line search we terminate with **status** = FAILURE if:

$$\alpha_P \leq \beta_3. \tag{33}$$

When this occurs we exit Algorithm 3 and go to line A.6 of Algorithm 1. On line A.6 we increases the size of δ and a new stabilization step is attempted. From Lemma 12 we know for sufficiently large δ the stabilization step will succeed.

To ensure that we do not perform unnecessary line searches we only attempt a stabilization line search if the following inequalities holds:

$$\tilde{\Delta}^{\phi_\mu}_{(x,y)}(d_x, d_y) < 0 \tag{34}$$

The idea of this equation is to only take steps when it is possible that ϕ can be decreased.

Algorithm 3 High level description of stabilization steps

Input: The point (x, y, s, μ), the matrix $\mathcal{K}_\delta$ and its factorization $\mathcal{K}_\delta^{-1}$.
An approximate log barrier hessian $\mathcal{M}$.
Output: A new point (x^+, y^+, s^+, μ^+) and a **status** of either SUCCESS or FAILURE

A.1 Compute the vector b at the point (x, y, s, μ) via (12) with $\eta = 0$.

A.2 Solve the system $\mathcal{K}_\delta d = -b$.

A.3 *Check that the direction has a reasonable chance of being accepted.* If (34) is not satisfied then terminate with **status** = FAILURE.

A.4 Estimate the largest primal step size $\alpha_P^{\max}$ from equation (25).

A.5 *Perform a backtracking line search on the primal step* α_P. Trial step sizes $\alpha_P \in \{\alpha_P^{\max}, \beta_6\alpha_P^{\max}, \beta_6^2\alpha_P^{\max}, ...\}$ computing the trial point (x^+, y^+, s^+, μ^+) as described in (16). Terminate with **status** = SUCCESS and return the trial point the first time all of the following conditions hold:

(i) The fraction to the boundary rule (17) is satisfied.

(ii) The set of valid dual step sizes is non-empty i.e. $B(s^+, d_y) \neq \emptyset$.

(iii) *Sufficient progress on filter.* Either equation (30) or (32) is satisfied.

Terminate with **status** = FAILURE if the step size becomes too small i.e. equation (33) is satisfied.

2.7 Algorithm Parameters

Table 1 Parameters values and descriptions

Parameter	Description	Possible values	Chosen value
$c_{\max}$	Maximum number of steps per outer iteration. See (A.4) of Algorithm 1.	Any natural number	3
β_1	Restricts how far complementarity of s and y can be from μ. See (18a).	The interval $(0, 1)$	0.01
β_2	Restricts how far complementarity of s and y can be from μ in order for the aggressive criterion to be met. See (24c).	The interval $(0, \beta_1)$	0.02
β_3	Minimum step size for stable line searches. See (33).	The interval $(0, 1)$	2^{-5}
β_4	Acceptable reduction factor for the scaled KKT error $\mathbb{K}_\mu$ during stabilization steps. See (32a).	The interval $(0, 1)$	0.2
β_5	Acceptable reduction factor for the merit function ϕ_μ during stabilization steps. See (30).	The interval $(0, 1)$	0.1
β_6	Backtracking factor for line searches in Algorithm 2 and 3.	The interval $(0, 1)$	0.5
β_7	Fraction to the boundary parameter.	The interval $(0, 1)$	0.01
β_8	Fraction to the boundary parameter used in (25) for computing the maximum step size $\alpha_P^{\max}$.	The interval $(\beta_7, 1)$	0.2
β_9	Exponent of $\|d_x\|$ used in fraction to boundary formula (25) for computing the maximum step size $\alpha_P^{\max}$.	The interval $(1, 2)$	1.5
β_{10}	Used in the regularizer defined in (6).	The interval $(0, \infty)$	10^{-8}
β_{11}	Used in the regularizer defined in (6).	The interval $(0, \infty)$	10^{-4}
$\delta_{\min}$		The interval $(0, \infty)$	10^{-8}
δ_{inc}		The interval $(1, \infty)$	8

3 Theoretical justification

The goal of this Section provide some simple theoretical justification for our algorithm. Section 3.1 justifies infeasibility termination criterion. Section 3.2 provides a simplified version of our algorithm, this is used to explain our algorithms global convergence properties. Section 3.3 proves that the algorithm described in Section 2 eventually converges.

3.1 Derivation of primal infeasibility termination criterion

The purpose of this section is to justify our choice of local infeasibility termination criterion, by showing that it corresponds to stationary measure for the infeasibility with respect to a weighted L_∞ norm. We also prove when the problem is convex our criterion certifies global infeasibility.

Consider the following optimization problem:

$$\min_x \max_i \bar{w}_i a_i(x) \tag{35a}$$

$$\text{s.t. } a_i(x) \leq 0, \forall i \in Z \tag{35b}$$

For some non-negative vector $\bar{w}$ with $\bar{w}_i = 0$ if and only if $i \in Z$. For example, a natural choice Z is the indices of constraints that have been chosen to be satisfied throughout the algorithm i.e. the bound

constraints and $\bar{w}_i = 1$ for $i \notin Z$. In this case, the problem reduces to

$$\begin{aligned} &\min_x \max_i a_i(x) \\ \text{s.t. } &a_i(x) \leq 0, \forall i \in Z \end{aligned}$$

Note that (35) is equivalent to the following optimization problem:

$$\min \mu \tag{36a}$$
$$\text{s.t. } a(x) + s = \mu w \tag{36b}$$
$$s, \mu \geq 0, \tag{36c}$$

where the vector w is defined by $w_i = \bar{w}_i^{-1}$ for $i \notin Z$ and $w_i = 0$ for $i \in Z$.

The KKT conditions for this problem are:

$$\begin{aligned} a(x) + s &= \mu w \\ \nabla a(x)^T \tilde{y} &= 0 \\ w^T \tilde{y} + u &= 1 \\ u\mu &= 0 \\ \tilde{y}^T s &= 0 \\ u, \mu &\geq 0. \end{aligned}$$

Note that if the point x, y, s, μ satisfies:

$$\begin{aligned} a(x) + s &= \mu w \\ \Gamma(x, y, s, \mu) &= 0 \\ \mu &> 0 \end{aligned}$$

then we have found an stationary point problem (35) with $\mu > 0$. In other words, a first order infeasibility certificate.

In the case, of convexity a stronger statement can be made:

Lemma 1. *Assume the constraint function a is convex and that some minimizer (x^*, μ^*) of (36) satisfies $\|x - x^*\|_2 \leq R$ for some constant $R > 0$. Suppose also that at some point (x, y, s, μ) one has:*

$$a(x) + s = \mu w \tag{37}$$
$$\Gamma(x, y, s, \mu) \leq \frac{1}{2m(R+1)}. \tag{38}$$

Then the system $a(x) \leq 0$ has no feasible solution.

Proof. Using $\tilde{y} := \frac{y}{\|Yw\|_\infty}$ we can write:

$$\begin{aligned} \tilde{y}^T a(x^*) &\geq \tilde{y}^T a(x) + \tilde{y}^T \nabla a(x)(x^* - x) \\ &= \tilde{y}^T (w\mu - s) + \tilde{y}^T \nabla a(x)(x^* - x) \\ &\geq \mu - m(1+R)\Gamma(x, y, s, \mu) \min\{1, \mu\} \\ &> \mu/2 \end{aligned}$$

The first transition holds via convexity, the second by (37), the third by the definition of Γ and the final inequality by (38). Therefore there exists no feasible solution to $a(x) \leq 0$. This is a typical Farkas infeasibility certificate argument. □

3.2 Global convergence proofs for a naive algorithm

Here we present Algorithm 4, a naive version of Algorithm 1. The goal of this naive algorithm is to illustrate the ideas of the global convergence proof of Algorithm 1. One should think of Algorithm 4 as mimicking the worst case performance of Algorithm 1. However, in practice, Algorithm 4 would be much slower than Algorithm 1. We emphasize that the goal of these convergence proofs is to prove asymptotic convergence to a stationary point, not to give a fast theoretical runtime bound. For work on fast theoretical runtimes for interior point methods with non-convex constraints see [REF OUR PAPER].

Algorithm 4 Naive version of Algorithm 1

Input: Some point x^0 and $\mu^0 > 0$ with $a(x^0) < \mu^0 e$

For $k = 0, ..., \infty$

A.1 *Stabilization stage.* Starting from x^{k-1} find any stationary point to the shifted log barrier problem $\min_x \psi_{\mu^k}(x)$ i.e.

$$x^k \in \{x \in \mathbb{R}^n : \nabla \psi_{\mu^k}(x) = 0\}$$

A.2 *Update dual and slack variables.* Set

$$s^k \leftarrow \mu^k e - a(x^k) \tag{39a}$$
$$y^k \leftarrow \mu^k (S^k)^{-1} e. \tag{39b}$$

A.3 *Check termination criterion.* If (20), (21) or (23) is satisfied then terminate the algorithm.

A.4 *Aggressive stage.* Set

$$\alpha_P^k \leftarrow \frac{1}{2} \min \left\{ 1, \frac{\min_i \{s_i^k\}}{\mu^k} \right\} \tag{40a}$$
$$\mu^{k+1} \leftarrow \mu^k (1 - \alpha_P^k) \tag{40b}$$
$$\bar{s}^k \leftarrow \mu^{k+1} - a(x^k) \tag{40c}$$
$$\bar{y}^k \leftarrow (\bar{S}^k)^{-1} \bar{\mu}^k. \tag{40d}$$

We now describe the similarities between Algorithm 1 and Algorithm 4. First note that the sequence:

$$(x^k, s^k, y^k, \mu^k) \tag{41}$$

corresponds to the subsequence satisfying the aggressive step criterion (24) in Algorithm 1. Furthermore, the sequence

$$(x^k, \bar{s}^k, \bar{y}^k, \mu^{k+1}) \tag{42}$$

corresponding to the subsequence of iterates generated by Algorithm 2 i.e. aggressive steps.

Now, we can describe the similarities between Algorithm 1 and Algorithm 4. In Algorithm 1, the stabilization step is repeatedly called until the aggressive step criterion (24) is met. This series of consecutive stabilization steps can be viewed as equivalent one call to the stabilization stage in Algorithm 4. Now, during an aggressive steps in Algorithm 1, if we let $\delta \to \infty$ then $x^+ \to x$. Hence (42) can be thought of as the 'worst case' aggressive sequence, corresponding to huge choice of δ. Now,

$$\bar{s}^k = \mu^{k+1} e - a(x^k) = \mu^{k+1} e - (\mu^k e - s^k) = s^k - \mu^k \alpha_P^k \geq s^k/2 > 0 \tag{43}$$

where the first equality follows by (40c), second by (39a), the third by (40b), and the first inequality from (40a). Hence each point x^k satisfies $\mu^{k+1} e - a(x^k) > 0$ and is therefore a strictly feasible starting point to the shifted log barrier problem $\psi_{\mu^{k+1}}(x)$ defined in line A.1 of Algorithm 4.

For simplicity of exposition, in Algorithm 4, we assume that we have some oracle that will find a stationary point of the sub-problem $\min_x \psi_{\mu^k}(x)$ given an initial point x^k. Most descent algorithms for unconstrained optimization will achieve this, assuming f and a are continuously differentiable. We show the convergence of the stabilization steps for this solving these sub-problems in Section 3.3.

In Theorem 2 we show this naive algorithm (Algorithm 4) eventually terminates. Here we sketch the proof. Assume, that at each x^k the first order local infeasibility is not satisfied (since otherwise the algorithm trivially terminates). A consequence of this assumption is that the dual variables are bounded from above. Since the dual variables are bounded above, we can bound the slack variables away from zero. By applying the update formula for μ^k given in (40b) we conclude μ^k is reduced sufficiently at each iteration and therefore $\mu^k \to 0$. Finally, observe that the iterates of Algorithm 4 satisfy

$$a(x^k) + s^k = e\mu^k \tag{44a}$$
$$\nabla \mathcal{L}(x^k, y^k) = 0 \tag{44b}$$
$$S^k y^k = \mu^k e \tag{44c}$$
$$s^k, y^k \geq 0 \tag{44d}$$

which is the central sequence property defined in equations (7) and (8), with $w = e$, as we discussed in the introduction. Hence, since $\mu^k \to 0$ eventually the optimality criterion is satisfied.

Theorem 2. *Assume that:*

- **A.** *The functions $f : \mathbb{R}^n \to \mathbb{R}$ and $a : \mathbb{R}^n \to \mathbb{R}^m$ are differentiable on $\mathbb{R}^n$.*
- **B.** *The input parameters satisfy $\epsilon_{opt} \in (0, 1)$.*
- **C.** *The level sets of the function $\psi_\mu(x)$ are bounded for all $\mu \leq \mu^0$.*
- **D.** *There exists some $G \geq \max\{\mu^0, \epsilon_{inf}\}$ such that $\|\nabla f(x^k)\|_\infty \leq G$.*

Then at any iteration k where the infeasibility termination criterion (21) *is not satisfied,*

$$\|y^k\|_\infty \leq \frac{2G}{\epsilon_{inf}\epsilon_{opt}}.$$

Furthermore, Algorithm 4 terminates in at most

$$1 + \frac{4G}{\epsilon_{opt}\epsilon_{inf}} \log\left(\frac{\mu^0}{\epsilon_{opt}}\right)$$

iterations.

Proof. First observe by the assumption that the level sets of $\psi_{\mu^k}(x)$ are bounded and the existence of a feasible solution x^k, the set $\{x \in \mathbb{R}^n : \nabla \psi_{\mu^k}(x) = 0\}$ is non-empty (since it contains $\arg\min_x \psi_{\mu^k}(x)$). Next, we have

$$\begin{aligned}
\|y^k\|_\infty &\leq \frac{\|\nabla a(x^k)^T y^k\|_\infty + \|Y^k s^k\|_\infty}{\epsilon_{\mathbf{inf}} \min\{1, \mu^k\}} \\
&\leq \frac{\|\nabla f(x^k)\|_\infty + \mu^k}{\epsilon_{\mathbf{inf}} \min\{1, \mu^k\}} \\
&\leq \frac{2G}{\epsilon_{\mathbf{inf}}\epsilon_{\mathbf{opt}}}
\end{aligned}$$

where the first inequality follows from the fact that the (21) is not satisfied; the second inequality from $\|\nabla \mathcal{L}(x^k, y^k)\|_\infty = 0$ and $S^k y^k = \mu^k$. The third from $G \geq \mu^k$ and $\epsilon_{\mathbf{opt}} \leq 1$. Now,

$$\begin{aligned}
\mu^{k+1} &\leq \mu^k - (1/2) \min_i s_i^k \\
&= \mu^k \left(1 - \frac{1}{2\|y^k\|_\infty}\right) \\
&\leq \mu^k \left(1 - \frac{\epsilon_{\mathbf{opt}}\epsilon_{\mathbf{inf}}}{4G}\right)
\end{aligned}$$

where the second equality holds from $S^k y^k = \mu^k e$, the second inequality by our bound on $\|y^k\|_\infty$. Noting that $\mu^k \leq \epsilon_{\mathbf{opt}}$ implies the algorithm terminates, gives the result. □

Keep in mind that Theorem 2 only bounds the number of iterations and excludes the computational cost of each solve of the stabilization stage.

3.3 Global convergence proofs for Algorithm 1

The majority of this section will be in the appendix for the real paper. The main result is Theorem 13 which shows global convergence of Algorithm 1 to a point satisfying the termination criterion.

3.3.1 Convergence of aggressive steps

The goal of this section is show that after a finite number of calls to aggressive steps Algorithm 1 converges. Lemma 3 does not rule out the possibility of an infinite sequence of failing aggressive steps.

Lemma 3. *Assume that:*

A. *The functions $f : \mathbb{R}^n \to \mathbb{R}$ and $a : \mathbb{R}^n \to \mathbb{R}^m$ are twice differentiable.*

B. *There exists some constant $G \geq \mu^0/\beta_1$ such that $\|\nabla f(x)\| \leq G$ for all $x \in \mathbb{R}^n$.*

C. *The tolerance $\epsilon_{\boldsymbol{opt}} \in (0, 1)$.*

Consider a point (x, y, s, μ) that satisfies the criterion for an aggressive step (24), but does not satisfy the infeasibility termination criterion (21) then:

$$\|Yw\|_\infty \leq \frac{2G}{\epsilon_{\boldsymbol{inf}}\epsilon_{\boldsymbol{opt}}}$$

*Furthermore, after a finite number of calls to Algorithm 2 that terminate with **status** = SUCCESS, Algorithm 1 will terminate at a point (x, y, s, μ) satisfying the local optimality termination criterion (20).*

Proof. We have

$$\begin{aligned}
\|Yw\|_\infty &\leq \frac{\max\{\|\nabla a(x)^T y\|_\infty, \|Sy\|_\infty\}}{\epsilon_{\mathbf{inf}} \min\{1, \mu\}} \\
&\leq \frac{\max\{\|\nabla f(x)\|_\infty + \|\nabla \mathcal{L}(x, y)\|_\infty, \mu/\beta_1\}}{\epsilon_{\mathbf{inf}} \min\{1, \mu\}} \\
&\leq \frac{2G}{\epsilon_{\mathbf{inf}}\epsilon_{\mathbf{opt}}}
\end{aligned}$$

where the first inequality follows from the fact that the (21) is not satisfied; the second inequality from the triangle inequality applied to $\|\nabla \mathcal{L}(x, y)\|_\infty$ and inequality (24c); and the third inequality from inequality (24b) and the Lemma's assumptions.

Next, for any trial step size α_P we have

$$\alpha_P \geq \min_{i \in N} \frac{\beta_6 s_i}{4\mu w_i} \geq \min_{i \in N} \frac{\beta_6 \beta_2}{4 w_i y_i} \geq \frac{\beta_6 \beta_2 \epsilon_{\mathbf{inf}}\epsilon_{\mathbf{opt}}}{8G}$$

where the first inequality is from the minimum trial step size from (26), the second inequality from (24c) and the final inequality from our bound on $\|Yw\|_\infty$.

Therefore we reduce μ by at least $\alpha_P \mu$ each iteration. Furthermore, for sufficiently small μ whenever (24) holds the optimality criterion (20) is satisfied. Combining these facts proves the Lemma. □

Lemma 4 shows that one can absorb the slack variable to reduce μ during aggressive steps by choosing a sufficiently large δ. Consequently, we are guaranteed if the criterion for an aggressive step is satisfied then there will be eventually an aggressive step taken with **status** = SUCCESS (since when an aggressive step fails for $j = 1$ the parameter δ is increased inside Algorithm 1 and eventually $\delta > \bar{\delta}$).

Lemma 4. *Let the functions $f : \mathbb{R}^n \to \mathbb{R}$ and $a : \mathbb{R}^n \to \mathbb{R}^m$ be twice differentiable. Consider an iterate (x, y, s, μ) that satisfies the criterion for an aggressive step (24) with $a(x) + s = \mu w$, then there exists some $\bar{\delta}$ such that for all $\delta > \bar{\delta}$, Algorithm 2 applied to the iterate (x, y, s, μ) terminates with* ***status*** = SUCCESS.

Proof. First, observe that as $\delta \to 0$ the direction d_x computed from (14) tends to zero. Consider any $\alpha_P \in (0, 1)$, since the function a is continuous for sufficiently large δ we have

$$\|a(x) - a(x + \alpha_P d_x)\|_\infty \leq \frac{1}{4} \min\{s_i\}$$

Assume to obtain a contradiction that Algorithm 2 fails. This implies by (26) that the backtracking line search the algorithm will have attempted some α_P such that

$$\alpha_P \in \left[0, \min_{i \in N} \frac{s_i}{4\mu w_i}\right] \tag{45}$$

then for this choice of α_P

$$\|s^+ - s\|_\infty = \| - \alpha_P \mu w + a(x) - a(x + \alpha_P d_x)\|_\infty \leq \frac{1}{2} \min\{s_i\}$$

where the first equality holds by (16c). Therefore $s^+ \in s[1/2, 3/2]$, hence the fraction to the boundary rule (17) is satisfied. Note that

$$\frac{s^+ Y}{\mu} \in [1/2, 3/2] \frac{sY}{\mu} \subseteq [\beta_2/2, 3/(2\beta_2)]e \subseteq [\beta_1, 1/\beta_1]e$$

therefore $\alpha_D = 0$ gives a feasible dual iterate. We obtain a contradiction, because the step should have been accepted. □

We conclude this subsection by summarizing the important aspects of the Lemma 3 and Lemma 4.

Lemma 5. *After a finite number of calls to Algorithm 2, Algorithm 1 terminates.*

However, Lemma 5 does not rule out the possibility there is an infinite number of consecutive stabilization steps. Ruling out this possibility is the purpose of subsection 3.3.2.

3.3.2 Convergence of stabilization steps

This subsection is devoted to showing that consecutive stabilization steps eventually satisfy the criterion for an aggressive step or the unboundedness criterion is satisfied. We now introduce $\mathbb{Q}_{\mu,C}$ which we will use to represent the set of possible points the iterates of Algorithm 1 can take for a fixed μ i.e. during consecutive stabilization steps.

Definition 6. *Define the set $\mathbb{Q}_{\mu,C}$ for constants $\mu, C > 0$ as the set of points $(x, y, s) \in \mathbb{R}^n \times \mathbb{R}^{m++} \times \mathbb{R}^{m++}$ such that*

A. *The function ϕ_μ is bounded above i.e. $\phi_\mu(x, y, s) \leq C$.*

B. *The unboundedness termination criterion (23) is not satisfied i.e.*

$$\frac{\| \max\{a(x), e\}\|_\infty}{\max\{1, -f(x)\}} \geq \epsilon_{\boldsymbol{unbd}}.$$

C. *The dual and slack variables are strictly positive i.e $y, s > 0$. Furthermore, equation (7a) and (7b) are satisfied:*

$$a(x) + s = \mu w$$
$$\frac{Sy}{\mu} \in [\beta_1 e, e/\beta_1]$$

Lemma 7. *Let the functions $f : \mathbb{R}^n \to \mathbb{R}$ and $a : \mathbb{R}^n \to \mathbb{R}^m$ be continuous. Then the set $\mathbb{Q}_{\mu,C}$ is compact.*

Proof. First consider the set

$$Q := \left\{x \in \mathbb{R}^n : (y,s) \in \mathbb{R}^{m++} \times \mathbb{R}^{m++}, \phi_\mu(x,y,s) \leq C, \| \max\{a(x), e\}\|_\infty \geq \max\{1, -f(x)\}\epsilon_{\mathbf{unbd}}\right\}$$

Now, consider some $(x,s) \in Q$. Recall that

$$\phi_\mu(x,y,s) = f(x) + \mu\left(\beta_{10}\sum_{i=1}^n \sqrt{x_i^2 + 1/\beta_{10}^2} - \beta_{11}e^T a(x)\right) - \mu\sum_i \log\left(\mu w_i - a_i(x)\right) + \frac{\|Sy-\mu\|_\infty^3}{\mu^2}.$$

The expression $\beta_{11}a_i(x) - \log(\mu w_i - a_i(x))$ and $\frac{\|Sy-\mu\|^3}{\mu^2}$ are bounded from bellow. Therefore there exists some constant $K_1 > 0$ such that

$$-K_1 \leq f(x) \leq K_1 - \beta_{10}\sum_i \sqrt{x_i^2 + 1/\beta_{10}^2}$$

It follows that x is bounded and therefore Q is bounded. Furthermore, since $\phi_\mu(x) \leq C$ and Q is bounded there exists some constant $K_2 > 0$ such that

$$a(x) \leq \mu w - K_2$$

for all $x \in Q$. Consider some sequence $x^k \in Q$ with $x^k \to x^*$. The statement $a(x) \leq \mu w - K_2$ implies ϕ_μ is continuous in a neighborhood of x^*. Using the definition of Q and the assumption that f and a are continuous implies $x^* \in Q$ i.e. Q is compact. Finally note that:

$$\mathbb{Q}_{\mu,C} = \left\{(x,y,s) \in \mathbb{R}^n \times \mathbb{R}^{m++} \times \mathbb{R}^{m++} : x \in Q, a(x) + s = \mu w, \frac{Sy}{\mu} \in [\beta_1 e, e/\beta_1]\right\}$$

is also compact. □

Corollary 8. *Let the functions $f : \mathbb{R}^n \to \mathbb{R}$ and $a : \mathbb{R}^n \to \mathbb{R}^m$ be twice differentiable. Then there exists some $L > 0$ such that for all $(x,y,s) \in \mathbb{Q}_{\mu,C}$ the following inequalities hold:*

$$s_i, y_i \geq 1/L$$

$$\|x\|, \|y\|, \|s\|, \|\nabla\psi_\mu(x)\|, \|\mathcal{M}\|, \|\nabla a(x)\| \leq L$$

and for any d s.t. $\|d\| < 1/L$

$$\psi_\mu(x+d) \leq \psi_\mu(x) - \nabla\psi_\mu(x)^T d + L\|d\|^2.$$

Furthermore, if the aggressive criterion (24) *does not holds:*

$$\max\{\|\nabla\psi_\mu(x)\|, \|Sy - \mu\|_\infty\} \geq 1/L$$

Proof. All these claims utilize the fact that an continuous function on a compact set is bounded above and bellow. □

With Corollary 8 in hand we proceed to showing that there will only be a finite number of stabilization steps until the next aggressive step.

Lemma 9. *Let the functions $f : \mathbb{R}^n \to \mathbb{R}$ and $a : \mathbb{R}^n \to \mathbb{R}^m$ be twice differentiable. There exists some $c_1, c_2, \bar{\delta} > 0$ such that for any $(x,y,s) \in \mathbb{Q}_{\mu,C}$, Algorithm 3 applied to the point (x,y,s,μ) for any $\delta \in [\bar{\delta}, \infty)$ terminates with **status** = SUCCESS using $\alpha_P = 1$ yielding a new point (x^+, y^+, s^+) such that:*

$$\psi_\mu(x^+) \leq \psi_\mu(x) + \beta_5\tilde{\Delta}_{(x,y)}^{\psi_\mu}(d_x) - c_1\|\nabla\psi_\mu(x)\|^2$$

Proof. Consider some $c_2 \in (\beta_5, 1)$ then there exists some constant $C > 0$ such that for all

$$\delta \geq C\left(\|\mathcal{M}\| + \|\nabla\psi_\mu(x)\|\right)$$

we have:

$$\lambda_{\min}/\lambda_{\max} \leq \sqrt{c_2}, 1 - 2L\lambda_{\max}/\lambda_{\min}^2 \geq \sqrt{c_2}, \|d_x\| < 1/L$$

where $\lambda_{\min}$ and $\lambda_{\max}$ denote the smallest and largest eigenvalues of the matrix $H = \mathcal{M} + \delta I$. It follows that

$$\begin{aligned}
\psi_\mu(x + d_x) - \psi_\mu(x) - \frac{\beta_5}{2} d_x^T \mathcal{M} d_x &\leq \nabla\psi_\mu(x)^T d_x + L\|d_x\|^2 - \frac{\beta_5}{2} d_x^T \mathcal{M} d_x \\
&\leq \nabla\psi_\mu(x)^T H^{-1} \nabla\psi_\mu(x) + 2L\|H^{-1}\nabla\psi_\mu(x)\|^2 \\
&\leq \|\nabla\psi_\mu(x)\|^2/\lambda_{\max}(1 - L\lambda_{\max}/\lambda_{\min}^2) \\
&\leq -c_2\|\nabla\psi_\mu(x)\|^2/\lambda_{\min}(H) \\
&\leq -c_2 \nabla\psi_\mu(x)^T H^{-1} \nabla\psi_\mu(x) \\
&= c_2 \nabla\psi_\mu(x)^T d_x \\
&\leq \beta_5 \nabla\psi_\mu(x)^T d_x - \frac{c_2 - \beta_5}{\lambda_{\max}} \|\nabla\psi_\mu(x)\|^2
\end{aligned}$$

□

Lemma 10. *Let the functions $f : \mathbb{R}^n \to \mathbb{R}$ and $a : \mathbb{R}^n \to \mathbb{R}^m$ be twice differentiable. For any $c_1 > 0$ there exists some $\bar{\delta} > 0$ such that for any $(x, y, s) \in \mathbb{Q}_{\mu,C}$, Algorithm 3 applied to the point (x, y, s, μ) for any $\delta \in [\bar{\delta}, \infty)$ yields a new point (x^+, y^+, s^+) with:*

$$\zeta(x^+, y^+) \leq \zeta(x, y) + \beta_5 \tilde{\Delta}^\zeta_{(x,y)}(d_x, d_y) + (c_1 - \zeta(x, y)(1 - \beta_5))$$

show that fraction to boundary rule is satisfied

Proof. Let $\gamma = \min\{(c_1/\mu)^{1/3}, 1\}$ and

$$\bar{\delta}_2(x) = \|\mathcal{M}\|_2 + \min\{\beta_8, \gamma\beta_1\} \frac{\|\nabla a(x)\|_\infty \|\nabla\psi_\mu(x)\|_\infty}{\min_i s_i}$$

then for any $\delta \geq \bar{\delta}_2(x)$ we have:

$$\begin{aligned}
\|S^{-1}d_s\|_\infty &\leq \frac{\|\nabla a(x)\|_\infty \|\nabla\psi_\mu(x)\|_\infty}{\delta - \|\mathcal{M}\|_2} \\
&\leq \min\{\beta_8, \gamma\beta_1\}
\end{aligned}$$

Hence the fraction to the boundary rule (25) and approximate complementarity (7b) are satisfied.

Now,

$$\begin{aligned}
\|Y^{-1}d_y\|_\infty &= \|\mu(YS)^{-1}e - e + S^{-1}d_s\|_\infty \\
&\leq (1/\beta_1 - 1) + \|S^{-1}d_s\|_\infty \leq 1/\beta_1
\end{aligned}$$

Hence

$$\|\mu - S^+ y^+\|_\infty \leq \mu\gamma$$

which implies

$$\zeta(x^+, y^+) \leq \mu^3\gamma^3/\mu^2 \leq c_1.$$

Since:

$$\zeta(x, y) + \tilde{\Delta}^\zeta_{(x,y)}(d_x, d_y) = 0$$

the result holds. □

Lemma 11. *Let the functions $f : \mathbb{R}^n \to \mathbb{R}$ and $a : \mathbb{R}^n \to \mathbb{R}^m$ be twice differentiable. There exists some $\bar{\delta} > 0$ such that for any $(x, y, s) \in \mathbb{Q}_{\mu,C}$, Algorithm 3 applied to the point (x, y, s, μ) with $\delta \geq \bar{\delta}$ terminates with **status** = SUCCESS.*

Proof. There exists some $c_2 > 0$ such that

$$c_2 - \zeta(x, y)/2 - c_1\|\nabla\psi_\mu(x)\|^2 \leq 0$$

for all $(x, y, s) \in \mathbb{Q}_{\mu,C}$. Combining Lemmas 9 and 10 we conclude that there exists some $\bar{\delta}$ such that for all $\delta \geq \bar{\delta}$

$$\tilde{\Delta}^{\phi_\mu}_{(x,y)}(u, v) = \tilde{\Delta}^{\psi_\mu}_{(x,y)}(u) + \tilde{\Delta}^{\zeta}_{(x,y)}(u, v). \tag{46}$$

□

Lemma 12. *After a finite number of consecutive stabilization steps either the aggressive criterion (24) or the unboundedness termination criterion (23) is met.*

Proof. Suppose in order to obtain a contradiction there is a infinite sequence of consecutive stabilization steps. Then there are two possibilities (i) there is an infinite sequence of consecutive steps where the sufficient decrease in the augmented log barrier function (30) is satisfied or (ii) there is an infinite subsequence of stabilization steps where a sufficient decrease in the KKT error (32a) is satisfied.

Since the unboundedness termination criterion (23) is not met, we know by Lemma 7 there exists some compact set $\mathbb{Q}_{\mu,C}$ that contains all the iterates.

Consider case (i). Consider any $\bar{\delta} > 0$ as defined in Lemma 11. Lemma 11 implies a step will be taken with $\delta \leq \delta_{\text{inc}}\bar{\delta}$. Now, consider any such step (x, y, s) to (x^+, y^+, s^+) with step size α_P. Using that the aggressive criterion (24) is not satisfied and that $\|\mathcal{M}(x, y, s, \mu)\|$ is bounded on the compact set $\mathbb{Q}_{\mu,C}$, we deduce there exists some constant $K > 0$ such that $\tilde{\Delta}^{\phi_\mu}_{(x,y)}(\alpha_P d_x, \alpha_P d_y) < K$ for all $\delta \leq \delta_{\text{inc}}\bar{\delta}$ and $\alpha_P > \beta_3$. moves a little to quick Since the criterion for sufficient progress on the augmented log barrier function (30) is satisfied, we deduce there exists some constant $K > 0$ such that $\phi_\mu(x^+, y^+, s^+) \leq \phi_\mu(x, y, s) - K$ at each iteration, which implies eventually the unboundedness criterion (23) is met.

Consider case (ii). Since (32a) holds for an infinite subsequence (x^k, y^k, s^k) and $\alpha_P > \beta_3$ by (33) we have $K_\mu(x^k, y^k, s^k) \to 0$. This implies eventually the aggressive criterion (24) is met.

Therefore neither case (i) or (ii) is possible. By contradiction the result holds. □

Theorem 13. *Algorithm 1 terminates after a finite number of computational operations.*

Proof. Lemma 5 show that the algorithm must terminate after a finite number of aggressive steps and each aggressive step occurs in a finite number of computational operations. Lemma 12 shows that the algorithm terminates or an aggressive step must be taken after a finite number of stabilization steps. The result follows. □

4 Implementation details

4.1 Initialization

This section explains how given a starting point x^0, how to select the initial variable values. The first goal is to modify x^0 such that it satisfies any bound constraints. This is done because often the non-linear constraints or objective may not be defined outside the bound constraints. Note that the way we have presenting our work the bound constraints are a subset of the set constraints given by $a_i(x)$ for $i = 1, ..., m$. We project onto the bounds in the same way as [Wächter and Biegler, 2006, Section 3.7].

The remainder of the intialization scheme is inspired by Mehrotra's work for linear programming [], but has been adapted to the non-linear programming context. We select a candidate dual variable and slack variables as follows

$$\tilde{y} \leftarrow \nabla a(x)(\nabla a(x)^T \nabla a(x) + I\kappa)^{-1} \nabla f(x) \tag{47}$$

$$\tilde{s} \leftarrow -a(x^0) \tag{48}$$

Consider the following scalar variables:

$$\varepsilon_y \leftarrow \max\{-2 \min_i y_i, 0\} \tag{49}$$

$$\varepsilon_s \leftarrow \max\left\{-2 \min_i s_i, \frac{\|\nabla \mathcal{L}(x^0, \tilde{y})\|_\infty}{\|\tilde{y}\|_\infty + 1}\right\} \tag{50}$$

then:

$$y^0 \leftarrow \tilde{y} + \varepsilon_y \tag{51}$$

$$s^0 \leftarrow \tilde{s} + \varepsilon_s \tag{52}$$

$$\mu^0 \leftarrow \frac{(s^0)^T y^0}{m} \tag{53}$$

Project μ^0 onto the interval $\|s\|_\infty[10^{-2}, 10^5]$. Project the dual variables y^0 onto the intervals:

$$\mu S^{-1} e[\beta_1, 1/\beta_1]$$

4.2 Linear algebra

explain how equality constraints can be handled

A. Splitting dense columns in sparse linear systems. Linear Algebra and its Applications. Robert J. Vanderbei. [Vanderbei, 1991]

B. [Lustig et al., 1991] Get between 5 times and 80 times speed up from splitting dense columns for stochastic programs.

C. Matrix Stretching for Sparse Least Squares `https://pdfs.semanticscholar.org/0054/9cc96c29f24c9d55d76962676pdf`

D. Matrix Stretching for Linear Equations `https://arxiv.org/abs/1203.2377`

E. J. F. Grcar, Matrix stretching for linear equations, Tech. Report SAND90-8723, Sandia National Laboratories, Nov. 1990.

4.3 Iterative refinement

5 Empirical results

For the empirical results we use the CUTEst non-linear programming test sets. The empirical results are structured as follows. Section 5.1 explores different algorithm options on CUTEst. Section 5.2 compares our algorithm against IPOPT on CUTEst. Section 5.3 compares our algorithm and IPOPT on a set of infeasible problems constructed from CUTEst.

For the comparisons we use IPOPT version 3.12.4 with the linear solver mumps. We turn off the nlp scaling, set the termination tolerance to 10^{-6}, set the boundary relaxation factor to zero. For both the one phase algorithm and IPOPT we set the time limit to 5 minutes and the maximum number of iterations $3,000$.

We selected a subset from the CUTEst test set of with more than 100 variables and constraints, but the total number of variables and constraints less than $10,000$. We further restricted the CUTEst problems to ones that are classified as having first and second derivatives defined everywhere (and available analytically). This gave us a test set with 238 problems.

5.1 Comparison of different algorithm options

Comparing algorithms in the following sections we use the performance profiling of [Dolan and Moré, 2002].

In Figure 1 we trial different line search conditions for the stabilization steps. In particular, we compare the default setting of a 'filter' as described in line (A.5) part (iii) of Algorithm 3 against other possible conditions. The first baseline to replace condition (iii) with (30) i.e. check that sufficient progress is made on the 'log barrier' merit function. The other baseline is removing condition (iii) entirely and simple taking the maximum step possible. Figure 1 indicates that the filter has superior performance of these three options.

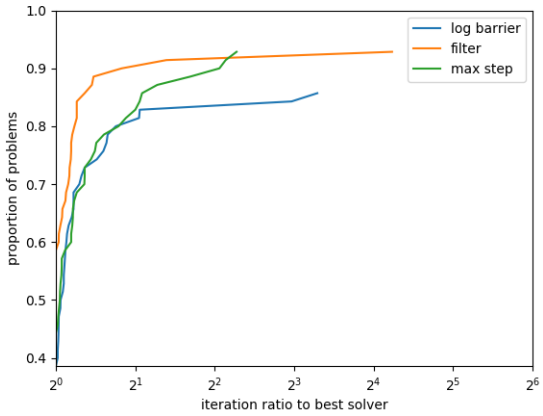

Figure 1: Comparison of different line search options.

In Figure 2 we compare different choices of the parameter $c_{\max}$, the maximum number of corrections ($c_{\max}$ is used on line A.4 in Algorithm 1). As one would expect, increasing the number of corrections decreases the iteration count, but has little impact on the failure rate. In the actual implementation of our one phase algorithm we chose $c_{\max} = 3$.

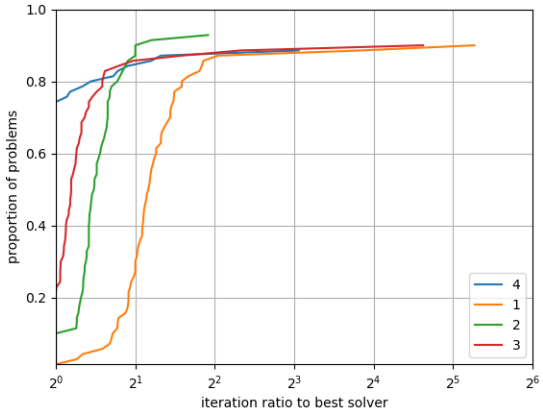

Figure 2: Comparison of the maximum number of corrections for the one phase algorithm.

5.2 Comparison with IPOPT

Please take these results with a grain of salt since we are comparing iteration counts not runtimes.

Comparison on number of function/constraint evaluations (clearly admit algorithm is not optimized to minimize constraint violation/function evaluation)

We consider the final function values f_*^a and f_*^b of algorithm a and b respectively approximately the same if:

$$\frac{f_*^a - f_*^b}{1 + \max\{|f_*^a|, |f_*^b|\}} < 10^{-1},$$

otherwise, we consider the solution of algorithm a better than algorithm b if $f_*^a < f_*^b$. For problems where both algorithm find a KKT point this is reported in the top three rows of Table 2. The remainder of Table 2 shows the number of times both algorithms succeed, fail, or just one algorithm fails. We consider the algorithm to have succeeded if it produces either a certificate of first order local optimality, infeasibility or unboundedness.

Let us highlight a few interesting facts from the tables. The one phase algorithm seem to find better KKT points on 13 problems versus 2 for IPOPT (Table 2). Furthermore, IPOPT fails on 39 problems compared with 21 problems for the one phase algorithm (Table 3). A large proportion of these failures occur before IPOPT has started (Table 4).

Table 2 Pairwise comparison of outcomes for IPOPT and the one phase algorithm

One Phase	IPOPT	#
same KKT	-	158
-	better KKT	2
better KKT	-	13
Succeed	Succeed	185
Fails	Fails	7
Succeed	Fails	32
Fails	Succeed	14

Table 3 Termination status counts

	One Phase	IPOPT
KKT	201	191
unbounded	4	0
primal infeasible	12	8
fail	21	39

Table 4 Failure reasons

	One Phase	IPOPT
max time	10	9
max iter	3	3
error before starting	4	19
error during algorithm	4	8
total	21	39

Figure 3 compares the iterations that IPOPT and the one phase algorithm take to succeed (produce a certificate of first order local optimality, infeasibility or unboundedness) on the CUTEst test set. Note that the iteration counts for the solvers are similar, except that the one phase solver fails less frequently.

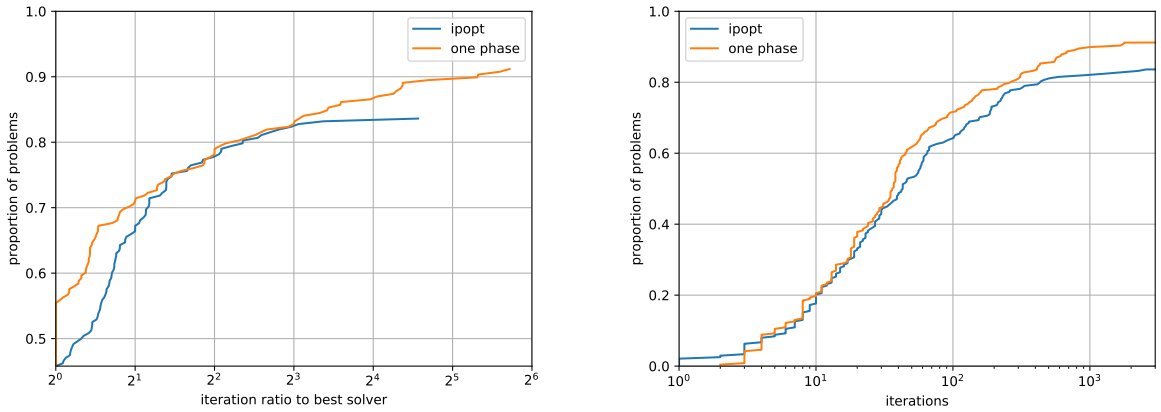

Figure 3: Comparison of IPOPT and one phase on CUTEst for problems where at least one solver declared the problem optimal, infeasible or unbounded.

table or plot of maximum dual variables?

5.3 Comparison on infeasible problems

Most of the CUTEst problems have feasible solutions. To generate a test set that was more likely to contain infeasible problems we perturbed the constraints as follows:

$$\tilde{a}(x) = a(x) + e$$

The solver terminated with the statuses described in the Table 5. This test was only run on problems with at most $1,000$ variables and constraints total.

Table 5 Termination status counts for perturbed CUTEst problems.

	One Phase	IPOPT
KKT	22	20
unbounded	1	0
primal infeasible	44	38
fail	3	12

Next, in Figure 4 we compare IPOPT and the one phase on the subset problems which at least one solver declared the problem locally infeasible. From this figure one can see that the one phase solver is quicker and more robust than IPOPT.

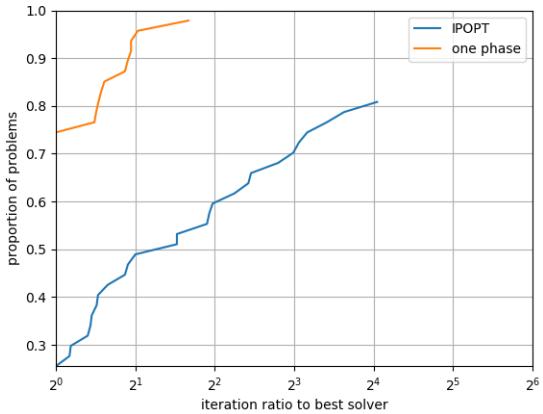

Figure 4: Comparison of IPOPT and one phase on perturbed CUTEst problems for which at least one solver declares the problem locally infeasible.

6 Conclusions

A. ??

7 To do

A. clean up

B. edit code to match document

C. run full CUTEst test

References

[Andersen and Ye, 1998] Andersen, E. D. and Ye, Y. (1998). A computational study of the homogeneous algorithm for large-scale convex optimization. *Computational Optimization and Applications*, 10(3):243–269.

[Andersen and Ye, 1999] Andersen, E. D. and Ye, Y. (1999). On a homogeneous algorithm for the monotone complementarity problem. *Mathematical Programming*, 84(2):375–399.

[Benson et al., 2004] Benson, H. Y., Shanno, D. F., and Vanderbei, R. J. (2004). Interior-point methods for nonconvex nonlinear programming: jamming and numerical testing. *Mathematical programming*, 99(1):35–48.

[Bian et al., 2015] Bian, W., Chen, X., and Ye, Y. (2015). Complexity analysis of interior point algorithms for non-lipschitz and nonconvex minimization. *Mathematical Programming*, 149(1-2):301–327.

[Byrd et al., 2006] Byrd, R. H., Nocedal, J., and Waltz, R. A. (2006). Knitro: An integrated package for nonlinear optimization. In *Large-scale nonlinear optimization*, pages 35–59. Springer.

[Chen and Goldfarb, 2006] Chen, L. and Goldfarb, D. (2006). Interior-point l2-penalty methods for nonlinear programming with strong global convergence properties. *Mathematical Programming*, 108(1):1–36.

[Curtis, 2012] Curtis, F. E. (2012). A penalty-interior-point algorithm for nonlinear constrained optimization. *Mathematical Programming Computation*, 4(2):181–209.

[Dolan and Moré, 2002] Dolan, E. D. and Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213.

[Fiacco and McCormick, 1990] Fiacco, A. V. and McCormick, G. P. (1990). *Nonlinear programming: sequential unconstrained minimization techniques*. SIAM.

[Fletcher and Leyffer, 2002] Fletcher, R. and Leyffer, S. (2002). Nonlinear programming without a penalty function. *Mathematical programming*, 91(2):239–269.

[Gabriel Haeser, 2017] Gabriel Haeser, Oliver Hinder, Y. Y. (2017). On the behavior of lagrange multipliers in convex and non-convex infeasible interior point methods. *arXiv*.

[Gould et al., 2015a] Gould, N. I., Orban, D., and Toint, P. L. (2015a). Cutest: a constrained and unconstrained testing environment with safe threads for mathematical optimization. *Computational Optimization and Applications*, 60(3):545–557.

[Gould et al., 2015b] Gould, N. I., Orban, D., and Toint, P. L. (2015b). An interior-point l1-penalty method for nonlinear optimization. *Numerical Analysis and Optimization*, pages 117–150.

[Huang and Mehrotra, 2016] Huang, K.-L. and Mehrotra, S. (2016). Solution of monotone complementarity and general convex programming problems using a modified potential reduction interior point method. *INFORMS Journal on Computing*, 29(1):36–53.

[Karmarkar, 1984] Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311. ACM.

[Kojima et al., 1989] Kojima, M., Mizuno, S., and Yoshise, A. (1989). A primal-dual interior point algorithm for linear programming. In *Progress in mathematical programming*, pages 29–47. Springer.

[Liu and Sun, 2004] Liu, X. and Sun, J. (2004). A robust primal-dual interior-point algorithm for nonlinear programs. *SIAM Journal on Optimization*, 14(4):1163–1186.

[Lustig, 1990] Lustig, I. J. (1990). Feasibility issues in a primal-dual interior-point method for linear programming. *Mathematical Programming*, 49(1-3):145–162.

[Lustig et al., 1991] Lustig, I. J., Mulvey, J. M., and Carpenter, T. J. (1991). Formulating two-stage stochastic programs for interior point methods. *Operations Research*, 39(5):757–770.

[McShane et al., 1989] McShane, K. A., Monma, C. L., and Shanno, D. (1989). An implementation of a primal-dual interior point method for linear programming. *ORSA Journal on computing*, 1(2):70–83.

[Megiddo, 1989] Megiddo, N. (1989). Pathways to the optimal set in linear programming. In *Progress in mathematical programming*, pages 131–158. Springer.

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

[Monteiro and Adler, 1989] Monteiro, R. D. and Adler, I. (1989). Interior path following primal-dual algorithms. part i: Linear programming. *Mathematical programming*, 44(1):27–41.

[Nocedal et al., 2014] Nocedal, J., Öztoprak, F., and Waltz, R. A. (2014). An interior point method for nonlinear programming with infeasibility detection capabilities. *Optimization Methods and Software*, 29(4):837–854.

[Shanno and Vanderbei, 2000] Shanno, D. F. and Vanderbei, R. J. (2000). Interior-point methods for nonconvex nonlinear programming: orderings and higher-order methods. *Mathematical Programming*, 87(2):303–316.

[Todd, 2003] Todd, M. J. (2003). Detecting infeasibility in infeasible-interior-point methods for optimization. Technical report, Cornell University Operations Research and Industrial Engineering.

[Vanderbei, 1991] Vanderbei, R. J. (1991). Splitting dense columns in sparse linear systems. *Linear Algebra and its Applications*, 152:107–117.

[Vanderbei, 1999] Vanderbei, R. J. (1999). Loqo user's manual—version 3.10. *Optimization methods and software*, 11(1-4):485–514.

[Wächter and Biegler, 2000] Wächter, A. and Biegler, L. T. (2000). Failure of global convergence for a class of interior point methods for nonlinear programming. *Mathematical Programming*, 88(3):565–574.

[Wächter and Biegler, 2005] Wächter, A. and Biegler, L. T. (2005). Line search filter methods for nonlinear programming: Motivation and global convergence. *SIAM Journal on Optimization*, 16(1):1–31.

[Wächter and Biegler, 2006] 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.

[Ye, 1998] Ye, Y. (1998). On the complexity of approximating a kkt point of quadratic programming. *Mathematical programming*, 80(2):195–211.

A Matrix factorization strategy

This strategy is based on the ideas of IPOPT [Wächter and Biegler, 2006, Algorithm IC].

Algorithm 5 Matrix factorization strategy

Input: The matrix $\mathcal{K}_0$ and current delta choice δ
Output: The factorization $\mathcal{K}_\delta^{-1}$ for some $\delta > 0$ such that the matrix $\mathcal{K}_\delta$ has the correct inertia.

A.1 Set $\delta_{\text{prev}} \leftarrow \delta$

A.2 Set $\delta \leftarrow 0$

A.3 Perform LDL factorization of $\mathcal{K}_\delta$, if inertia is correct return $\mathcal{K}_\delta^{-1}$ otherwise continue.

A.4 If $\delta_{prev} > 0$ set $\delta \leftarrow \max\{\delta_{\min}, \delta_{\text{prev}}/3\}$ otherwise set $\delta = \delta_{\text{start}}\mu$.

A.5 Perform LDL factorization of $\mathcal{K}_\delta$, if inertia is correct return $\mathcal{K}_\delta^{-1}$ otherwise continue.

A.6 Set $\delta \leftarrow 8\delta$. Go to previous step.

B The (non-existence) of a central path in non-convex optimization

Would be nice to have a long discussion on this issue

$$f_\mu(x) = 50(x - 0.5)^3 + x - \mu(\log(x) + \log(1 - x))$$

$$\nabla f_\mu(x) = 150.0 * (x - 0.5)^2 + 1.0 - \mu/x + \mu/(1 - x) = 0$$

Is discontinuous at $\mu = 3$, $x \approx 0.5$ i.e. there exists no function $x(\mu)$ such that $\nabla f_\mu(x(\mu)) = 0$ and $x(\mu)$ is continuous.

[Vanderbei' s example for the problem $\min x - x^2$ s.t. $x \geq 0$ there exists no continuous central path from an initial point to the optimal solution. However, optimal solution is unbounded.]

C Old

C.1 Intuition: penalty method versus infeasible start method

Give a simple example illustrating the draw backs of a penalty method

A. Makes the algorithm more complex

B. If penalty parameter is too big then problem is harder to solve than it should be

C. When penalty parameter is updated the dual feasibility increases suddenly

C.2 Discussion of Watcher and Biegler's example

Key differences:

A. Non-linear updates

B. Initialization of slack variables violates their assumptions

$$x_1^2 + a \geq 0 \tag{54}$$
$$x_1 \geq b \tag{55}$$

$$\min -\mu(x_1^2 + a) - (x_1 - b) \tag{56}$$
$$w \geq 0 \tag{57}$$

At

$$\mu = 1/(2x_1)$$

C.3 Relevant literature

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

A. The paper [Wächter and Biegler, 2000] 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 [Wächter and Biegler, 2005]. 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 [Byrd et al., 2006]. 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 [Nocedal et al., 2014] that adds an feasibility restoration phase (this is theoretically unnecessary, but the practical results are good).

D. The paper [Curtis, 2012] 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?

F. why homogenous algorithm fails: relies on KKT conditions to measure progress

C.4 Old convergence proofs

I keep on re-writing these as the algorithm changes, so the current proofs are not up to date. Will revise these once the algorithm stabilizes.

Lemma 14. *Consider Algorithm 1. Assume that the slack variables are initialized such that* $s^1 \leftarrow \mu^1 w - a(x^1)$ *for some* $\mu^1, w \geq 0$ *such that* $s^1 > 0$*. If the criterion for an aggressive step* (24) *is met at any point during the algorithm then for the current dual variable* y *we have:*

$$\|y^k\|_1 \leq \frac{\|\nabla f(x^k)\|_2}{\epsilon_{\boldsymbol{inf}}^2} + 3m$$

$$w^T y^k \leq \frac{\|\nabla f(x^k)\|_2 + \mu(1 + \|W\|_\infty)}{\mu^k \epsilon_{\boldsymbol{inf}}}$$

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:

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

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:

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

Then:

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

Which gives the result. □

Lemma 15. *Consider Algorithm 1. Assume that the slack variables are initialized such that* $s^1 \leftarrow \mu^1 w - a(x^1)$ *for some* $\mu^1, w \geq 0$ *such that* $s^1 > 0$*. Algorithm 1 takes at most* $\frac{\mu^0(2\|\nabla c(x^k)\|_2 + 8)}{\epsilon^2}$ *aggressive steps to satisfy the termination criterion i.e satisfy* (20)*,* (21) *or* (23)*.*

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

$$\alpha \leq \frac{1}{\|y^k\|_\infty + 4} \tag{58}$$

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. □