A one phase IPM for non-convex optimization

A one phase interior point method (IPM) for non-convex optimization

Oliver Hinder, Yinyu Ye

Stanford University
ohinder@stanford.edu

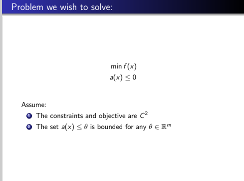
October 14, 2016

(slow, no ums, time) Hey, my name is Oliver and today I will be talking about work in progress with Yinyu Ye on developing a one phase method for non-convex optimization.

The purpose of this work is to develop an interior point algorithm for non-convex optimization that is simpler than existing approaches, but still has excellent convergence properties and practical performance.

This talk is split into three sections.

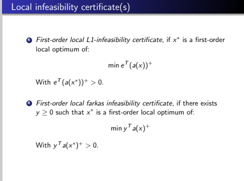
1. In first section, we review current non-convex IPM and explain why one phase algorithms are currently not used for non-convex optimization. We then outline our algorithm.
2. In the second section, we present our convergence results notably, we do not require constraint qualifications for either global or local super-linear convergence.
3. In the third sections, we present preliminary CUTEst results.



So the problem we want to solve has general non-linear objective and inequality constraints, we can, of course, represent any equality constraint by two inequalities.

We assume that:

1. The functions have 1st and 2nd derivatives everywhere
2. The feasible region is bounded for any perturbation of the right hand side. If necessary, one can add dummy constraints to guarantee this. This assumption simplifies the analysis.



- Unlike convex in non-convex $\rightarrow$ cannot make global guarantee of infeasibility $\rightarrow$ need local guarantee
- Multiple local inf. cert. - different ways of measuring tot. violation.
    1. (slide) L1 or any other norm e.g. L2, L3.
        - If second-order sufficient $\rightarrow$ local minimizer
        - typical non-convex
    2. (slide) Farkas
        - Used in convex
        - L1 $\rightarrow$ Farkas (weaker)

While there are other possible many other possible criterion we could use, in this talk we will focus on these two.

Either declare problem infeasible (primal or dual) or optimal in one phase:

- Homogenous algorithm [?, ?]
- Infeasible start IPM [?, ?, ?]

- One phase successful in LP/conic.
- Starting from an infeasible point → optimal or infeasible

Two one phase algorithm worth mentioning:

1. HA was developed for LP by Ye in 1994 and extended to general convex in 1999. The HA prove infeasibility and unboundedness by convergence of the iterates. While the HA is ideal for convex optimization its reliance on the existence of a central path makes it difficult to adapt to non-convex optimization.
2. Infeasible start IPM for linear programming was first suggested in 1990 by Lustig then perfected by Mehotra in 1992. As Todd showed in 2002, the infeasible start algorithm proves infeasibility for LP through divergence of the iterates. The infeasible start algorithm is the basis of this talk.

2016-10-14

Failure of infeasible start IPM for non-convex optimization

[?] 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

1. If infeasible start algorithms are so effective, why are they are not used for non-convex optimization?
2. As [**?**] showed, for a range of starting points ...
3. Strongly indicates that the infeasible start algorithm IPM needs to be significantly modified to guarantee convergence ...

2016-10-14

1. Two phases (IPOPT)
2. Compute two directions (KNITRO)
3. Penalty (or big-M) method e.g. [?, ?]

Following this paper there were countless different papers proposing very significant modifications to the infeasible start IPM to guarantee convergence.

I wish to briefly discuss three of these.

1. Two phases (IPOPT)
2. Compute two directions (KNITRO)
3. Penalty (or big-M) method e.g. [?, ?]

Two phases (IPOPT):

- Main phase
- Feasibility restoration phase

Criticism:

- Each phase has a different variables: each time initialize FRP start from scratch, the dual variables = zero, new introduced primal variables have ad hoc values.
- Awkward convergence assumptions (to prove convergence it is assumed constraints are linearly independent in a neighborhood of the feasible region)
- Poor performance on infeasible problems.

1. Two phases (IPOPT)
2. Compute two directions (KNITRO)
3. Penalty (or big-M) method e.g. [?, ?]

(KNITRO) Compute two directions

- One direction aims to minimize L2 norm of constraint violation, other step aims to improve optimality
- To compute directions need to factorize two different linear systems at each step.

Modifications to the infeasible start IPM

1. Two phases (IPOPT)
2. Compute two directions (KNITRO)
3. Penalty (or big-M) method e.g. [?, ?]

Penalty:

- Another approach is to add the constraints into the objective using a penalty parameter
- ... algorithm becomes more complicated - **simultaneously** adjust a barrier μ^k penalty parameter and a feasibility penalty parameter

... Now, as you can see each of these approaches requires non-trivial modifications to the infeasible start IPM to guarantee convergence.

And the take away point of this talk is:

It doesn't need to be this complicated!

(9 minutes) It doesn't need to be this complicated!

In this talk, I will show despite the fact an infeasible start algorithm fails with equality constraints, if we:

1. Use inequality constraints instead equality constraints
2. Use a non-linear update for the slack variables
3. Adjust the target rate of reduction of constraint violation and the barrier parameter

Then an infeasible start IPM can be proved to converge to a farkas certificate of infeasibility or local optimality certificate.

Furthermore, if we add dummy constraints, this farkas certificate, becomes a certificate of L1 minimization of the constraints.

One phase barrier problem

$$\min f(x)$$
$$a(x) \leq 0$$

1. Add a slack variable
2. Add a log barrier term
3. Keep the constraint violation the same
4. Reduce the constraint violation and μ^k by $\eta^k \in [0, 1]$
5. Add proximal term

Now we are ready to start describing our algorithm. Recall the original problem we wish to solve with a general non-linear objective and inequality constraints.

One phase barrier problem

$$\min f(x)$$
$$a(x) + s = 0$$
$$s \geq 0$$

1. Add a slack variable
2. Add a log barrier term
3. Keep the constraint violation the same
4. Reduce the constraint violation and μ^k by $\eta^k \in [0, 1]$
5. Add proximal term

One phase barrier problem

$$\min f(x) - \mu^k \sum_i \log s_i$$

$$a(x) + s = 0$$

$$s \geq 0$$

1. Add a slack variable
2. Add a log barrier term
3. Keep the constraint violation the same
4. Reduce the constraint violation and μ^k by $\eta^k \in [0, 1]$
5. Add proximal term

2016-10-14

A one phase IPM for non-convex optimization

└Introduction

└Algorithm outline

└One phase barrier problem

$$\min f(x) - \mu^k \sum_i \log s_i$$
$$a(x) + s = (a(x^k) + s^k)$$
$$s \geq 0$$

1. Add a slack variable
2. Add a log barrier term
3. Keep the constraint violation the same
4. Reduce the constraint violation and μ^k by $\eta^k \in [0, 1]$
5. Add proximal term

One phase barrier problem

$$\min f(x) - (1-\eta^k)\mu^k \sum_i \log s_i$$
$$a(x) + s = (1-\eta^k)(a(x^k) + s^k)$$
$$s \geq 0$$

- Add a slack variable
- Add a log barrier term
- Keep the constraint violation the same
- Reduce the constraint violation and μ^k by $\eta^k \in [0,1]$
- Add proximal term

One phase barrier problem

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

1. Add a slack variable
2. Add a log barrier term
3. Keep the constraint violation the same
4. Reduce the constraint violation and μ^k by $\eta^k \in [0,1]$
5. Add proximal term

[pause] So, now we have our barrier sub-problem, we need to derive a primal-dual direction. To do this we ...

Primal-dual direction computation

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

1. Form KKT system for barrier problem
2. Linearize the KKT system
3. Factorize matrix and compute direction d^k

Primal-dual direction computation

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

$$H^k = \nabla_x^2 L(x^k, y^k) + \delta^k I$$

1. Form KKT system for barrier problem
2. Linearize the KKT system
3. Factorize matrix and compute direction d^k

Primal-dual direction computation

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

$$H^k = \nabla_x^2 L(x^k, y^k) + \delta^k I$$

1. Form KKT system for barrier problem
2. Linearize the KKT system
3. Factorize matrix and compute direction d^k

Iterate update

For step size $\alpha^k \in [0,1]$ update iterates as follows:

$$
\begin{aligned}
x^{k+1} &= x^k + \alpha^k d_x^k \\
y^{k+1} &= y^k + \alpha^k d_y^k \\
s^{k+1} &= (1 - \alpha^k \eta^k)(a(x^k) + s^k) - a(x^{k+1}) \\
\mu^{k+1} &= (1 - \alpha^k \eta^k)\mu^k
\end{aligned}
$$

Only accept iterates that approximately satisfy complementary:

$$\beta \mu^{k+1} \leq s_i^{k+1} y_i^{k+1} - \mu^{k+1} \leq \frac{1}{\beta}\mu^{k+1}$$

For some constant $\beta \in (0,1)$.

1. We update the x and y variables in a using a linear update
2. The s variables are updated with a non-linear update to ensure the constraint violation is reduced by **exactly** α^k times η^k.
3. Furthermore, we reject any iterates that do not approximately satisfy complementary.

One phase non-convex algorithm

for $k \leftarrow 1, \ldots, \infty$ **do**

if $\|\nabla f(x^k) + (y^k)^T \nabla a(x^k)\| < \mu^k$ **then**
Aggressive step:
Compute direction with $\eta^k \in (0, 1]$
Take largest step while maintaining complementary
else
Stabilization step:
Compute direction with $\eta^k = 0$
Backtracking line search on direction using merit function
end if

end for

On this slide we present an outline of our algorithm. At each iteration, we either take an aggressive step or a stabilization step.

1. Aggressive step (are we approximately opt. to sub-problem)
   - In direction computation we set $\eta^k \in (0, 1]$ to simultaneously reduce the constraint violation and μ.
   - Take largest possible step
2. Stabilization step:
   - In direction computation we set $\eta^k = 0$ to keep the constraint violation and μ exactly the same.
   - We then perform a backtracking line search on a log barrier merit function.

Termination criterion

First-order ϵ-locally optimal if:

$$\left\| \nabla f(x) + y^T \nabla a(x) \right\| + \|a(x) + s\| + \mu < \epsilon$$

First-order ϵ-locally farkas infeasible if:

$$\frac{\mu + \|y^T \nabla a(x)\|}{y^T (a(x) + s)} < \epsilon$$

There are two ways our algorithm can terminate.

2016-10-14

Why poly-time is hard for interior point methods

Suppose we are given a feasible starting point to the following problem:

$$\min_x f(x) - \mu^k \sum_i \log(-a(x))$$

- Log barrier is $O(1/\delta^2)$ Lipschitz continuous on the set $\{x : a(x) \leq -\delta\}$ for any $\delta > 0$
- Bounding δ using the log barrier merit function:

$$\delta = \Omega\left(\exp(-1/\mu^k)\right)$$

Implies exponential runtime in μ^k for gradient descent or newton's method!!!

Global poly-time convergence proof sketch

Prove that there exists $\delta^k = O(\|y^k\|)$ such that:

$$\delta^k \|d_x^k\| = O\left(1 + \sqrt{\|y^k\| \mu^k}\right)$$

i.e. independent of r.h.s. This implies:

$$\|KKT^k\| = O\left(1 + \sqrt{\|y^k\| \mu^k}\right)$$

Which implies assuming the algorithm has not produced an infeasibility certificate that:

$$s_i^k = \Omega((\mu^k \epsilon)^2)$$

Global poly-time convergence

1. Differentiability and boundedness (defined earlier)
2. Slack variable s^0 is chosen carefully e.g. such that $a_i(x^0) + s_i^0 > 0$
3. That δ^k and η^k are judiciously chosen

Theorem

If (1-3) hold, then for any $\epsilon > 0$, after at most $\tilde{O}(\epsilon^{-17})$ steps the algorithm has found either a first-order ϵ-local optimum or a first-order ϵ-local farkas infeasibility certificate.

(15 minutes) Note:

1. No constraint qualifications required
2. When we are converging to a local optimum this does not guarantee the sequence of dual multipliers is bounded

Note:

1. No constraint qualifications required - this is first infeasible start IPM with this property
2. No guarantee if we are converging to infeasibility certificate

First-order L1-infeasibility certificate

$$\min f(x)$$
$$a(x) \leq 0$$

This problem is trivially equivalent to:

$$\min f(x)$$
$$a(x) \leq z$$
$$e^T z \leq 0$$
$$z \geq 0$$

Farkas infeasibility certificate $\rightarrow$ L1-infeasibility certificate

So far we have shown that if our algorithm does not find a local optimum it finds a farkas certificate of local infeasibility.

What if we would like a different type of infeasibility certificate? For example, if we want to minimize the sum of constraint violation then it is sufficient to embed the original problem inside a slightly larger problem.

Assuming the slack variables are appropriately initialized then the farkas certificate of local infeasibility for the larger problem will imply that we have minimized the sum of constraint violation.

2016-10-14

Table: Results on 30 selected CUTEst problems, number of variables between 100 and 1000

	One phase	IPOPT (2004)	KNITRO (2004)
Median #iterations	71 (59)	41	53
SEGFAULTS	4	0	0
MAXIT	3	3	3

1. On this slide we compare the iteration count for a preliminary implementation of our algorithm.
2. The median number of iterations for our algorithm is 71 compared with 41 for IPOPT and 53 for KNITRO
3. There is much work still needed for our implementation, but these results at least indicate that we are competitive.

1. Infeasible start IPM only needs minor modifications to guarantee convergence
2. For global and super-linear convergence: no constraint qualifications
3. Early computational results promising - more work needed

- Believed lit infeasible start IPM significantly modified to guarantee convergence
- We:
  - Use inequality constraints instead equality constraints
  - Use a non-linear update for the slack variables
  - Adjust the target rate of reduction of constraint violation and the barrier parameter

  Then an infeasible start IPM can be proved to converge to a farkas certificate of infeasibility or local optimality certificate.
- Furthermore, if we add dummy constraints, this farkas certificate, becomes a certificate of L1-infeasibility certificate.

2016-10-14

A one phase IPM for non-convex optimization

The End

A one phase IPM for non-convex optimization

2016-10-14

References

References I

Andreas Wächter, Lorenz T. Biegler (2000)
Failure of global convergence for a class of interior point methods for nonlinear programming. *Mathematical Programming*.

Yinyu Ye, Michael J. Todd, and Shinji Mizuno (1994)
An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm *Mathematics of Operations Research*.

Erling D. Andersen and Yinyu Ye (1999)
On a homogeneous algorithm for the monotone complementarity problem. *Mathematical Programming*.

Irvin J Lustig (1990)
Feasibility issues in a primal-dual interior-point method for linear programming. *Mathematical Programming*.

Mehrotra Sanjay (1992)
On the implementation of a primal-dual interior point method. *SIAM Journal on optimization*.

2016-10-14

A one phase IPM for non-convex optimization

└─References

Michael J. Todd (2002)
Detecting infeasibility in infeasible-interior-point methods for optimization.
Foundations of Computational Mathematics.

Frank E. Curtis (2012)
A penalty-interior-point algorithm for nonlinear constrained optimization.
Mathematical Programming Computation.

Lifeng Chen and Donald Goldfarb (2006)
A penalty-interior-point algorithm for nonlinear constrained optimization.
Mathematical Programming.