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

Oliver Hinder, Yinyu Ye

ohinder@stanford.edu

July 17, 2016

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

Oliver Hinder, Yinyu Ye

ohinder@stanford.edu

July 17, 2016

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.

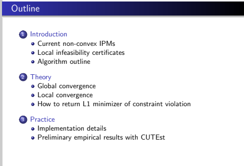
Currently, in optimization literature it is thought that infeasible start IPM needed to be significantly modified in order to guarantee convergence for non-convex optimization problems. In this talk, we demonstrate by avoiding using equality constraints and gradually reducing the constraint violation it is possible to give convergence guarantees for a pure infeasible start algorithm. Furthermore, preliminary computational tests indicate this could become efficient algorithm in practice.

Outline

1. Introduction
   - Current non-convex IPMs
   - Local infeasibility certificates
   - Algorithm outline
2. Theory
   - Global convergence
   - Local convergence
   - How to return L1 minimizer of constraint violation
3. Practice
   - Implementation details
   - Preliminary empirical results with CUTEst

A one phase IPM for non-convex optimization

Outline



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 superlinear convergence.
3. In the third sections, we briefly discuss implementation details and present preliminary CUTEst results.

Problem we wish to solve:

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

Assume:

1. The constraints and objective are C^2
2. The set $a(x) \leq \theta$ is bounded for any $\theta \in \mathbb{R}^m$

Problem we wish to solve:

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

Assume:

1. The constraints and objective are C^2
2. The set $a(x) \leq \theta$ is bounded for any $\theta \in \mathbb{R}^m$

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.

One phase IPMs for convex optimization

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

1. Infeasible start IPM (Mehortra, 1992)
2. Homogenous algorithm (Anderson and Ye, 1998, 1999)

State of the art for convex optimization

One phase IPMs for convex optimization

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

1. Infeasible start IPM (Mehortra, 1992)
2. Homogenous algorithm (Anderson and Ye, 1998, 1999)

State of the art for convex optimization

One phase algorithms have been successful in linear programming and convex optimization. Starting from an infeasible point they declare a problem optimal or infeasible. At each iteration one fixed newton system is used and the right hand side to decide the target reduction in primal, dual and complementary to get quick convergence.

The two types most successful one phase algorithms are the infeasible start IPM that proves infeasibility through the divergence of dual iterates and the homogenous algorithm that proves infeasibility by convergence.

These algorithms have been state of the art for convex optimization, both in terms of theory and practical performance.

Why one phase non-convex IPM do not exist (yet)

1. Some 'pure' infeasible start algorithms have been tested, but no theoretical guarantees (i.e. LOQO)
2. Ideally our algorithm would either find a local optimum or a local minimizer the violation of the constraints
3. Infeasible start IPMs may fail this criterion (Wachter and Biegler in 2000).

Why one phase non-convex IPM do not exist (yet)

1. Some 'pure' infeasible start algorithms have been tested, but no theoretical guarantees (i.e. LOQO)
2. Ideally our algorithm would either find a local optimum or a local minimizer the violation of the constraints
3. Infeasible start IPMs may fail this criterion (Wachter and Biegler in 2000).

1. Some 'pure' infeasible start algorithms have been tested, but no theoretical guarantees (i.e. LOQO)
2. Ideally our algorithm would either find a local optimum or a local minimizer of the constraint violation
3. In 2000, Wachter and Biegler gave an example of a problem for which infeasible start algorithm may fail to converge. I think this paper influenced many IPMs including IPOPT and KNITRO.

Solutions in literature to this problem

1. Two phases (IPOPT)
    1. Each phase has a different variables
    2. Awkward convergence assumptions
    3. Poor performance on infeasible problems
2. Compute two directions (KNITRO)
3. Penalty (or big-M) method (Curtis, 2012)

Solutions in literature to this problem

1. Two phases (IPOPT)
   1. Each phase has a different variables
   2. Awkward convergence assumptions
   3. Poor performance on infeasible problems
2. Compute two directions (KNITRO)
3. Penalty (or big-M) method (Curtis, 2012)

1. Two phases (IPOPT): main phase and feasibility restoration phase
   - 1.1 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.
   - 1.2 Awkward convergence assumptions: to prove convergence it is assumed constraints are linearly independent in a neighborhood of the feasible region.
   - 1.3 Poor performance on infeasible problems.
2. (KNITRO) using two different newton systems
   - 2.1 One direction aims to minimize L2 norm of constraint violation, other step aims to improve optimality
   - 2.2 Not ‘pure’ IPM in feasibility search, requires factorizing two different newton systems at each step.

Solutions in literature to this problem

1. Two phases (IPOPT)
    1. Each phase has a different variables
    2. Awkward convergence assumptions
    3. Poor performance on infeasible problems
2. Compute two directions (KNITRO)
3. Penalty (or big-M) method (Curtis, 2012)

- Two phases (IPOPT)
  - Each phase has a different variables
  - Awkward convergence assumptions
  - Poor performance on infeasible problems
- Compute two directions (KNITRO)
- Penalty (or big-M) method (Curtis, 2012)

1. Curtis use L1 minimization

... Now, as you can see there are a lot of complicated modifications to the infeasible start IPM to guarantee convergence.

It doesn't need to be this complicated!

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. Adjust the target rate of reduction of primal and complementary

Then an infeasible start algorithm 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.

Local infeasibility certificates

Farkas certificate, there exists $y \geq 0$ such that x^* is a stationary point of:

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

With $y^T a(x^*) > 0$.

L1 constraint violation, x^* is a stationary point of:

$$\min e^T (a(x))^+$$

With $e^T (a(x^*))^+ > 0$.

Local infeasibility certificates

Farkas certificate, there exists $y \geq 0$ such that x^* is a stationary point of:

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

With $y^T a(x^*) > 0$.

L1 constraint violation, x^* is a stationary point of:

$$\min e^T (a(x))^+$$

With $e^T (a(x^*))^+ > 0$.

In convex optimization infeasibility detection is simple. We produce a farkas certificate, that proves the problem is infeasible. In non-convex optimization we cannot guarantee global infeasibility. However, there are many possible types of local infeasibility certificates.
For example, we could produce a farkas certificate of infeasibility, which is a non-negative weight vector y and decision variable x^* such that x^* is a stationary point of $y^T a(x)$ with $y^T a(x^*)$ positive.
A stronger infeasibility criterion would be if x^* was a stationary point of the sum of the constraint violation.

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

One phase barrier problem

Original problem:

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

Barrier problem:

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

One phase barrier problem

Original problem:

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

Barrier problem:

$$
\begin{aligned}
&\min f(x) - (1-\eta^k)\mu^k \sum_i \log s_i\\
&a(x) + s = (1-\eta^k)(a(x^k) + s^k)\\
&s \ge 0
\end{aligned}
$$

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.
Our idea is to modify this problem in two ways. Firstly, we add in a log barrier term to penalize getting too close to the boundary of the constraints, this is a very typical strategy. Secondly, we set a target for reducing the constraint violation. It is this second feature, which is common in convex codes, but uncommon in non-convex codes, which is critical to our result.
At each iteration we reduce the penalty parameter μ^k and the primal feasibility by the same rate η^k.

Primal-dual direction computation

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

$$
\begin{aligned}
H^k &= \nabla_x^2 L(x^k, y^k) + \delta^k I \\
A^k &= \nabla a(x^k)
\end{aligned}
$$

Primal-dual direction computation

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

$$
\begin{aligned}
H^k &= \nabla_x^2 L(x^k, y^k) + \delta^k I \\
A^k &= \nabla a(x^k)
\end{aligned}
$$

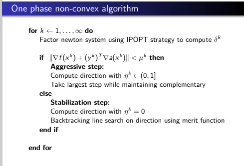
Next, we take the barrier problem and derive a typical primal-dual interior point direction. We modify the inertia of the hessian matrix to ensure it is sufficiently positive definite. Observe that by changing η^k which only appears in the right hand side we can change the target reduction in primal feasibility and duality gap.

One phase non-convex algorithm

for $k \leftarrow 1, \ldots, \infty$ **do**
  Factor newton system using IPOPT strategy to compute δ^k

  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.

If the dual feasibility is sufficiently small, then we take an aggressive step. In an aggressive step we reduce μ^k and the primal feasibility by η^k. We then take the largest step in that direction while maintaining complementary.

Otherwise, we take a stabilization step where we keep the primal feasibility and complementary the same. We perform a backtracking search on the direction using a merit function.

Iterate update

Backtrack on α and update iterates:

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

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 $\beta \in (0, 1)$.

Iterate update

Backtrack on α and update iterates:

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

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 $\beta \in (0, 1)$.

Given a direction, and a step size α^k this slide describes how we update the iterates. We update the x and y variables in a typical fashion, however, the s variables and barrier parameter μ are updated to ensure that the constraint violation and μ are reduced by **exactly** α^k times η^k.

Furthermore, we reject any iterates that do not approximately satisfy complementary.

Merit function line search

Backtrack on α until armijo rule is satisfied for the merit function:

$$\psi^k(x) = f(x) - \mu^k \sum_i \log\left(a_i(x^k) + s_i^k - a_i(x)\right)$$

Merit function line search

Backtrack on α until armijo rule is satisfied for the merit function:

$$\psi^k(x) = f(x) - \mu^k \sum_i \log\left(a_i(x^k) + s_i^k - a_i(x)\right)$$

During the stabilization step, we use the following, typical, barrier merit function to ensure convergence. We use a backtracking line search to take the largest step that makes sufficient progress on this merit function.

During the aggressive step, we do not use a merit function.

Termination criterion

Declare problem first-order ϵ-locally optimal if:

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

Declare problem first-order ϵ-locally (farkas) infeasible if:

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

Termination criterion

Declare problem first-order ϵ-locally optimal if:

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

Declare problem first-order ϵ-locally (farkas) infeasible if:

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

Finally, there are two ways our algorithm can terminate.

1. It satisfies the KKT conditions to sufficient degree of accuracy
2. It finds a Farkas certificate of infeasibility

Global convergence

If δ^k and η^k are judiciously chosen then it possible to show the following result:

Theorem

For any $\epsilon > 0$, after a finite number of steps the algorithm has found either a first-order ϵ-local optimum or a first-order ϵ-local (farkas) infeasibility certificate.

Global convergence

If δ^k and η^k are judiciously chosen then it possible to show the following result:

Theorem

For any $\epsilon > 0$, after a finite number of steps the algorithm has found either a first-order ϵ-local optimum or a first-order ϵ-local (farkas) infeasibility certificate.

Now, assuming δ^k and η^k are carefully chosen then it possible to show that our algorithm either converges is a first-order ϵ-local optimum or a first-order ϵ-local farkas infeasibility certificate. It is worth noting that this result does not say anything about whether the sequence of dual multipliers is bounded.

Local convergence

Assume:

1. Converging towards a KKT solution that satisfies strict complementary.
2. The sufficient conditions for local optimality are satisfied at the KKT solution.

If η^k is judiciously chosen then it is possible to show:

Theorem

If (1-2) hold, the algorithm converges Q-quadratically to a local optimality certificate.

Local convergence

Assume:

1. Converging towards a KKT solution that satisfies strict complementary.
2. The sufficient conditions for local optimality are satisfied at the KKT solution.

If η^k is judiciously chosen then it is possible to show:

Theorem

If (1-2) hold, the algorithm converges Q-quadratically to a local optimality certificate.

Our one phase algorithm has nice convergence properties. If the algorithm is converging towards a KKT point that satisfies strict complementary and the sufficient conditions for local optimality are satisfied at this point, then the algorithm converges Q-quadratically to a local optimality certificate. Observe that this result is somewhat unusual since we require minimal assumptions and no constraint qualifications.

It is worth noting this result holds because we take the largest step while maintaining complementary during the aggressive phase.

How to return L1 minimizer of constraint violation

We wish to solve:

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

This problem is trivially equivalent to:

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

The local Farkas infeasibility certificate to this problem is an certificate of L1 minimization of the constraint violation.

How to return L1 minimizer of constraint violation

We wish to solve:

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

This problem is trivially equivalent to:

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

The local Farkas infeasibility certificate to this problem is an certificate of L1 minimization of the constraint violation.

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.

Implementation details

1. Use fraction to boundary rule i.e. $y^{k+1} \geq 0.05 y^k$ and $s^{k+1} \geq 0.05 s^k$
2. Choose η^k on aggressive steps using predictor-corrector technique
3. Search on negative eigenvector to guarantee second-order necessary conditions are satisfied

Implementation details

1. Use fraction to boundary rule i.e. $y^{k+1} \geq 0.05 y^k$ and $s^{k+1} \geq 0.05 s^k$
2. Choose η^k on aggressive steps using predictor-corrector technique
3. Search on negative eigenvector to guarantee second-order necessary conditions are satisfied

Our implemented algorithm, is more sophisticated than the simple algorithm described at the beginning of the talk. We are in the early stages and the design decisions are still fluid. I am just going to briefly discuss a few implementation details:

1. To prevent iterates use converging too quickly to the boundary, we only accept iterates that satisfy the fraction to the boundary criterion.
2. For the theoretical proofs we never give a practical choice for η^k, in practice, we use a predictor-corrector technique. First, we compute take an aggressive step with $\eta^k = 1$, and on aggressive steps using predictor-corrector technique
3. We use the factorized matrix as a pre-conditioner to generate negative eigenvectors. We search on this negative eigenvector to guarantee second-order necessary conditions are satisfied.

Scaled KKT merit function

Scaled KKT residuals is an alternate merit function:

$$\frac{\|\nabla f(x) + y^T \nabla a(x)\|}{\|y\|}$$

Accept step that makes progress on either:

1. Classic merit function
2. Scaled KKT residuals

Using a filter to prevent cycling

Scaled KKT merit function

Scaled KKT residuals is an alternate merit function:

$$\frac{\|\nabla f(x) + y^T \nabla a(x)\|}{\|y\|}$$

Accept step that makes progress on either:

1. Classic merit function
2. Scaled KKT residuals

Using a filter to prevent cycling

We have found that the practical performance of the algorithm can be improved substantially by adding a second merit function during the stabilization step.

Empirical results

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

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

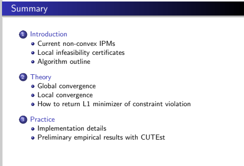
Table: Results on 30 selected CUTEst problems, number of variables between 100 and 1000

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

On this slide we compare the iteration count for an old version of our algorithm compared with IPOPT and KNITRO. This algorithm is based on similar ideas, but has significant differences from the current implementation.

Summary

1. Introduction
   - Current non-convex IPMs
   - Local infeasibility certificates
   - Algorithm outline
2. Theory
   - Global convergence
   - Local convergence
   - How to return L1 minimizer of constraint violation
3. Practice
   - Implementation details
   - Preliminary empirical results with CUTEst



Finally, before the presentation ends, I wish to give a quick summary. Currently, in optimization literature it is thought that infeasible start IPM needed to be significantly modified in order to guarantee convergence for non-convex optimization problems. In this talk, we have demonstrated by avoiding using equality constraints and gradually reducing the constraint violation it is possible to give convergence guarantees for a pure infeasible start algorithm. Furthermore, preliminary computational tests indicate this could become efficient algorithm in practice.

The End

The End

References

Itai Ashlagi, Mark Braverman, and Avinatan Hassidim (2014)
Stability in Large Matching Markets with Complementarities
Operations Research 62:4, 713-732.

References

Itai Ashlagi, Mark Braverman, and Avinatan Hassidim (2014)
Stability in Large Matching Markets with Complementarities
Operations Research 62:4, 713-732.

Conjectures

Conjecture

Assume (x^*, s^*) is a limit point of the algorithm and there exists finite dual multipliers at (x^*, s^*) then for some subsequence of the dual multipliers $y^{\pi(k)} \to y^*$.

Conjecture

After a polynomial number of steps a simple variant of the algorithm has found either a first-order ϵ-local optimum or a first-order ϵ-local infeasibility certificate.

Conjectures

Conjecture

Assume (x^*, s^*) is a limit point of the algorithm and there exists finite dual multipliers at (x^*, s^*) then for some subsequence of the dual multipliers $y^{\pi(k)} \to y^*$.

Conjecture

After a polynomial number of steps a simple variant of the algorithm has found either a first-order ϵ-local optimum or a first-order ϵ-local infeasibility certificate.