Infeasibility Detection in Nonlinear Optimization

Frank E. Curtis, Lehigh University
Hao Wang, Lehigh University

SIAM Conference on Optimization 2011

May 16, 2011

(See also "Infeasibility Detection in Nonlinear Programming," Figen Oztoprak, MS71)

Outline

Outline

Motivation

Penalty-SQO Method

Penalty-Interior-Point Method

Unified Framework(?)

Constrained minimization

Is this how we should formulate optimization problems?

$$\min_{x} f(x) \quad \text{s.t.} \begin{cases} c^{\mathcal{E}}(x) = 0 \\ c^{\mathcal{I}}(x) \leq 0 \end{cases} \tag{OP}$$

Feasibility violation minimization

Suppose that if (OP) is infeasible, then we want to solve

$$\min_x \; v(x) := \left\| \begin{bmatrix} c^{\mathcal{E}}(x) \\ \max\{c^{\mathcal{I}}(x), 0\} \end{bmatrix} \right\| \tag{FP}$$

Many algorithms/codes do this already, by either

- switching from solving one problem to the other;
- transitioning from solving one problem to the other.

But are they doing it efficiently?

Numerical experiments: Infeasible optimization problems

Iterations and f evaluations for 8 infeasible optimization problems (2-3 variables):

Prob.	Ipopt		Knitro-Direct		Knitro-Active		Filter	
	Iter.	Eval.	Iter.	Eval.	Iter.	Eval.	Iter.	Eval.
1	48	281	38	135	22	235	16	16
2	109	170	*10000	*40544	23	167	12	12
3	788	3129	12	83	9	202	10	10
4	46	105	25	61	10	201	11	11
5	72	266	*1060	*3401	18	45	26	26
6	63	141	*76	*264	16	37	27	27
7	87	152	*10000	*43652	*10000	*20091	30	30
8	104	206	33	97	41	560	28	28

Problems also run with `Knitro-CG` and `LOQO`, but they failed every time.

(We want your infeasible test problems!)

Numerical experiments: Feasibility problems (solved directly)

Iterations and f evaluations for 8 feasibility problems (2-3 variables):

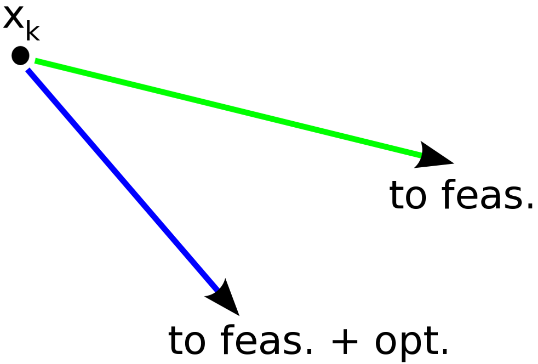
Problem	Ipopt		Knitro-Direct		Knitro-Active		Filter	
	Iter.	Eval.	Iter.	Eval.	Iter.	Eval.	Iter.	Eval.
1	28	29	14	15	13	24	17	21
2	31	32	31	33	8	9	12	13
3	50	131	10	11	9	12	12	13
4	24	79	18	29	10	13	10	12
5	166	786	29	40	21	24	30	32
6	37	48	20	21	19	22	26	27
7	59	65	31	34	19	20	25	28
8	46	71	19	20	23	28	26	29

$\Longrightarrow$ If we can switch/transition efficiently, then our current tools work well.

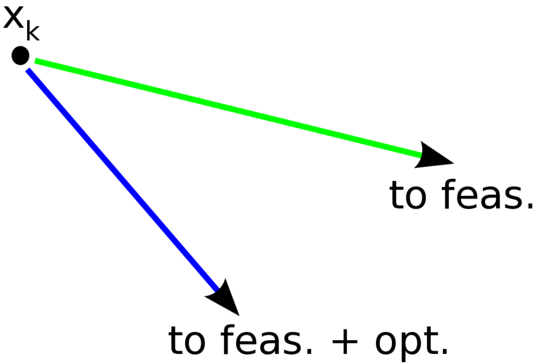
Rapid detection of infeasibility

Problem type	Global convergence	Fast local convergence
Feasible	✓	✓
Infeasible	✓	?

Two-phase strategy



Two-phase strategy



- Exploratory step determines possible progress toward feasibility.
- Rapid infeasibility detection requires rapid transition/switch to feasibility steps.
- (Important to have consistency between ways the two steps measure feasibility.)

Outline

Motivation

Penalty-SQO Method

Penalty-Interior-Point Method

Unified Framework(?)

Penalty methods

Penalization provides one mechanism for transitioning to infeasibility detection:

$$\min_{x,r,s,t} \rho f(x) + e^T r + e^T s + e^T t$$
$$\text{s.t.} \begin{cases} c^{\mathcal{E}}(x) = r - s \\ c^{\mathcal{I}}(x) \leq t \\ (r, s, t) \geq 0 \end{cases} \qquad \text{(PP)}$$

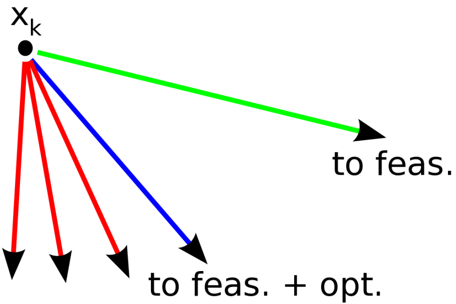
How should ρ be updated?

Literature

- (Rich history of SQP methods)
- Fletcher, Leyffer (2002)
- Byrd, Gould, Nocedal (2005)
- Byrd, Nocedal, Waltz (2008)
- Byrd, Curtis, Nocedal (2010)
- Byrd, López-Calva, Nocedal (2010)
- Gould, Robinson (2010)
- Morales, Nocedal, Wu (2010)

Steering and FILTER strategies

- Steering methods solve a sequence of constrained subproblems:



- FILTER: "... we make use of a property of the phase I algorithm in our QP solver. If an infeasible QP is detected, the solver exists with a solution of [an LP minimizing the ℓ_1 norm of violated constraints]."

SQuID strategy

- Compute v_k as the optimal value for

$$\min_{d,r,s,t} \; e^T r + e^T s + e^T t$$
$$\text{s.t.} \begin{cases} c^{\mathcal{E}}(x_k) + \nabla c^{\mathcal{E}}(x_k)^T d = & r - s \\ c^{\mathcal{I}}(x_k) + \nabla c^{\mathcal{I}}(x_k)^T d \leq & t \\ \|d\|_\infty \leq & \Delta \\ (r,s,t) \geq & 0 \end{cases}$$

- Update ρ for rapid infeasibility detection (Byrd, Curtis, Nocedal, 2010)
- Compute d_k as the search direction from

$$\min_{d,r,s,t} \; \rho(f(x_k) + \nabla f(x_k)^T d) + e^T r + e^T s + e^T t + \tfrac{1}{2} d^T H(x_k, \lambda_k; \rho) d$$
$$\text{s.t.} \begin{cases} c^{\mathcal{E}}(x_k) + \nabla c^{\mathcal{E}}(x_k)^T d = & r - s \\ c^{\mathcal{I}}(x_k) + \nabla c^{\mathcal{I}}(x_k)^T d \leq & t \\ e^T r + e^T s + e^T t \leq & v_k \\ (r,s,t) \geq & 0 \end{cases}$$

- Update ρ for descent

Numerical experiments: Infeasible optimization problems

Iterations and f evaluations for 8 infeasible optimization problems (2-3 variables):

Prob.	Filter		SQuID	
	Iter.	Eval.	Iter.	Eval.
1	16	16	32	65
2	12	12	12	18
3	10	10	14	36
4	11	11	26	57
5	26	26	5	16
6	27	27	18	18
7	30	30	19	19
8	28	28	17	17

Outline

Penalty and interior-point methods

Constrained subproblems in penalty methods can be expensive:

$$\min_{x,r,s,t} \rho f(x) + e^T r + e^T s + e^T t \quad \text{s.t.} \begin{cases} c^{\mathcal{E}}(x) = r - s \\ c^{\mathcal{I}}(x) \leq t \\ (r,s,t) \geq 0 \end{cases} \tag{PP}$$

Interior-point methods are more efficient for large-scale problems:

$$\min_{x,u} f(x) - \mu \sum \ln u^i \quad \text{s.t.} \begin{cases} c^{\mathcal{E}}(x) = 0 \\ c^{\mathcal{I}}(x) = -u \\ u \geq 0 \end{cases} \tag{IP}$$

Penalty-interior-point method

Applying an interior-point reformulation to (PP):

$$\min_{x,r,s,t,u} \rho f(x) - \mu \left(\sum (\ln r^i + \ln s^i) + \sum (\ln t^i + \ln u^i) \right) + e^T r + e^T s + e^T t$$
$$\text{s.t.} \begin{cases} c^{\mathcal{E}}(x) = & r - s \\ c^{\mathcal{I}}(x) = & t - u \end{cases} \tag{PIP}$$

The optimization problem (OP) and feasibility problem (FP) can be solved via (PIP):

- $\mu \to 0$ and $\rho \to \bar{\rho} > 0$ to solve (OP).
- $\mu \to 0$ and $\rho \to 0$ to solve (FP).

Literature

Previous work with similar motivations:

- Jittorntrum and Osborne (1980)
- Polyak (1982, 1992, 2008)
- Breitfeld and Shanno (1994, 1996)
- Goldfarb, Polyak, Scheinberg, and Yuzefovich (1999)
- Gould, Orban, and Toint (2003)
- Chen and Goldfarb (2006, 2006)
- Benson, Sen, and Shanno (2008)

Literature

Previous work with similar motivations:

- Jittorntrum and Osborne (1980)
- Polyak (1982, 1992, 2008)
- Breitfeld and Shanno (1994, 1996)
- Goldfarb, Polyak, Scheinberg, and Yuzefovich (1999)
- Gould, Orban, and Toint (2003)
- Chen and Goldfarb (2006, 2006)
- Benson, Sen, and Shanno (2008)

Parameter updates are essential to have a practical algorithm.

Conservative strategy:

1. Fix $\rho > 0$ and (approximately) solve (PIP) for $\mu \to 0$.
2. If infeasible, then reduce ρ and go to step 1.

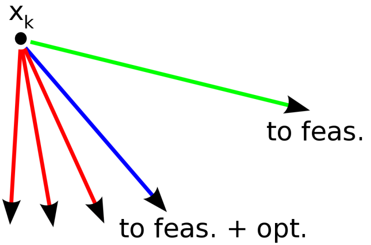
(ρ reduced too slowly to ensure fast convergence for infeasible problems.)

PIPAL strategy

Newton system has the form

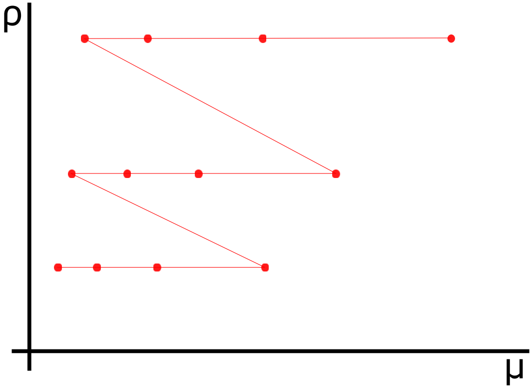
$$A_{\rho,\mu} d = \rho q_1 + \mu q_2 + q_3.$$

Steering strategy now efficient! (Only one factorization required.)



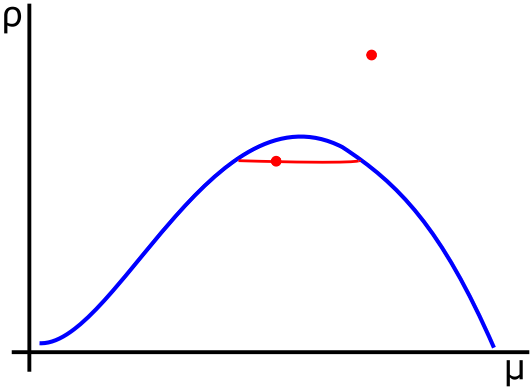
- Solution for $(\rho, \mu) = (0, \mu)$ indicates progress possible toward primal feasibility.
- Adjust ρ to aim for primal feasibility.
- Adjust μ to aim for dual feasibility and complementarity.

Conservative strategy



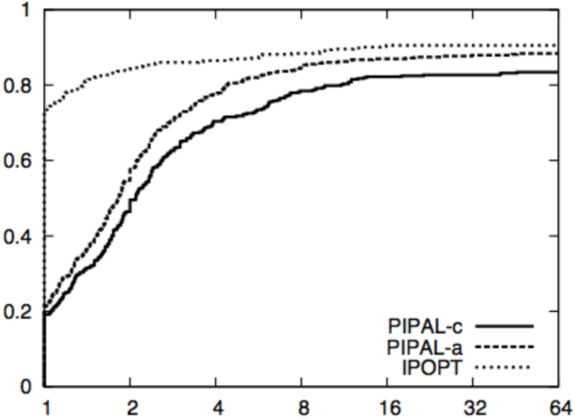
- Each “dot” may require at least a few iterations.
- Each “row” may require the computational effort of an entire interior-point solve.
- For an infeasible problem, we need both ρ and μ to reduce to zero.

PIPAL strategy

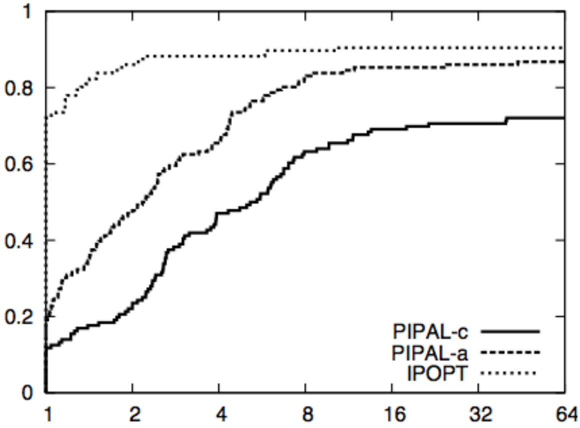


- Blue line separates acceptable and unacceptable ρ values (for primal feasibility).
- μ then set to ensure progress toward dual feasibility and complementarity.
- For an infeasible problem, ρ and μ can reduce rapidly.

Numerical results: Feasible problems (sample size = 422)

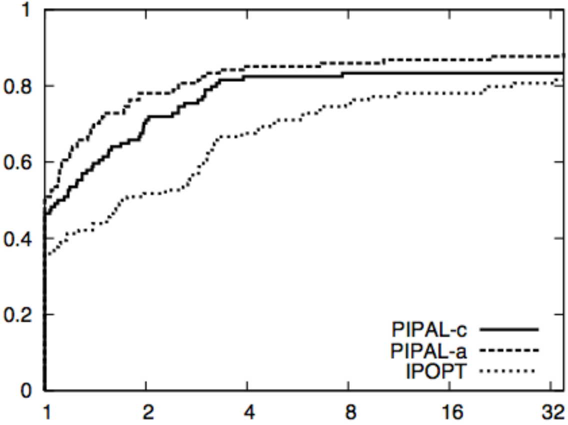


Numerical results: Feasible problems w/ ρ decrease (sample size = 136)



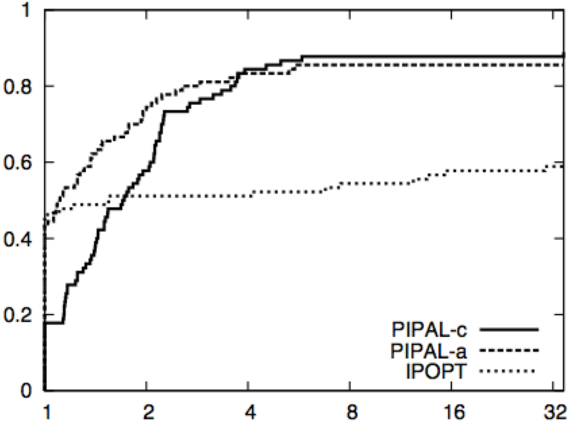
Numerical results: Degenerate problems (sample size = 114)

Added constraints: $c^i(x)^2 \geq 0$



Numerical results: Infeasible problems (sample size = 90)

Added constraints: $c^i(x)^2 \leq -1$



Outline

Constrained minimization

Is this how we should formulate optimization problems?

$$\min_x \; f(x) \quad \text{s.t.} \begin{cases} c^{\mathcal{E}}(x) = 0 \\ c^{\mathcal{I}}(x) \leq 0 \end{cases} \tag{OP}$$

That is, should the strategy be the following:

- Attempt to solve (OP), but...
- if it is infeasible, then solve

$$\min_x \; v(x) := \left\| \begin{bmatrix} c^{\mathcal{E}}(x) \\ \max\{c^{\mathcal{I}}(x), 0\} \end{bmatrix} \right\| \tag{FP}$$

Minimization over set of feasibility violation minimizers

Let $\mathcal{X}$ be the solution set of feasibility violation minimization:

$$\min_x \; v(x) := \left\| \begin{bmatrix} c^{\mathcal{E}}(x) \\ \max\{c^{\mathcal{I}}(x), 0\} \end{bmatrix} \right\| \tag{FP}$$

We should solve

$$\begin{aligned} \min_x \; & f(x) \\ \text{s.t. } & x \in \mathcal{X}. \end{aligned} \tag{UP}$$

(Even better would be a single problem or set of conditions for solving (UP).)

$1 \leq 0$

Consider the following model (AMPL presolver turned off):

```
var      x;
minimize f: (x-1)^2;
s.t.     c: 1 <= 0;
```

Solver	Inf. Declared?	Iterations	Final (x, y)
CONOPT	No	0	0.0000
FILTER	Yes	0	0.0000
IPOPT	Yes	6	0.0099
KNITRO	Yes	7	0.6667
LANCELOT	Yes	1	1.0000
LOQO	No	500	0.6728
MINOS	Yes	0	0.0000
MOSEK	Yes	1	1.0000
PENNON	No	0	1.0000
SNOPT	Yes	0	0.0000

$y^2 \leq -1$

Consider the following model (AMPL presolver turned off):

```
var      x;
var      y;
minimize f: (x-1)^2;
s.t.     c: y^2 + 1 <= 0;
```

Solver	Inf. Declared?	Iterations	Final (x, y)
CONOPT	Yes	3	(0.0000,0.0000)
FILTER	Yes	2	(0.0000,0.0000)
IPOPT	Yes	6	(0.0099,0.0000)
KNITRO	Yes	4	(0.3765,0.0000)
LANCELOT	Yes	1	(1.0000,0.0000)
LOQO	No	500	(0.4569,0.0000)
MINOS	Yes	2	(1.0000,0.0000)
MOSEK	Yes	1	(0.0000,0.0000)
PENNON	No	0	(1.0000,0.0000)
SNOPT	Yes	3	(1.0000,0.0000)

Penalty and switching methods

- Penalization provides one mechanism for transitioning to infeasibility detection:

$$\min_{x,r,s,t} \ \rho f(x) + e^T r + e^T s + e^T t$$
$$\text{s.t.} \begin{cases} c^{\mathcal{E}}(x) = & r - s \\ c^{\mathcal{I}}(x) \leq & t \\ (r,s,t) \geq & 0 \end{cases} \tag{PP}$$

However, once $x \in \mathcal{X}$ is found, $f(x)$ has been thrown out (since $\rho \to 0$).

- Switching methods throw out $f(x)$ even earlier.

What is needed?

General-purpose algorithms that:

- require no constraint qualifications to do something useful
- minimize the objective over the set of feasibility violation minimizers
- are as efficient and robust as contemporary methods when problems are nice