Recent Adaptive Methods for Nonlinear Optimization

Frank E. Curtis, Lehigh University

involving joint work with

James V. Burke (U. of Washington), **Richard H. Byrd** (U. of Colorado), **Nicholas I. M. Gould** (RAL), **Zheng Han** (American Express), **Johannes Huber** (SAFEmine), **Hao Jiang** (Johns Hopkins), **Travis Johnson** (Square, Inc.), **Jorge Nocedal** (Northwestern), **Daniel P. Robinson** (Johns Hopkins), **Olaf Schenk** (U. della Svizzera Italiana), **Philippe L. Toint**, (U. of Namur), **Andreas Wächter** (Northwestern), **Hao Wang** (ShanghaiTech), **Jiashan Wang** (U. of Washington)

ExxonMobil Research — Annandale, NJ

8 July 2015

Outline

Outline

Constrained nonlinear optimization

Consider the constrained nonlinear optimization problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \ & f(x) \\ \text{s.t. } & c(x) \leq 0, \end{aligned} \tag{NLP}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ are continuously differentiable.

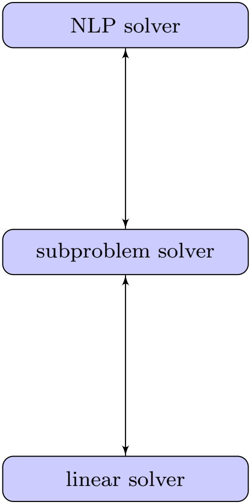
(Equality constraints also OK, but suppressed for simplicity.)

We are interested in algorithms such that if (NLP) is infeasible, then there will be an automatic transition to solving the feasibility problem

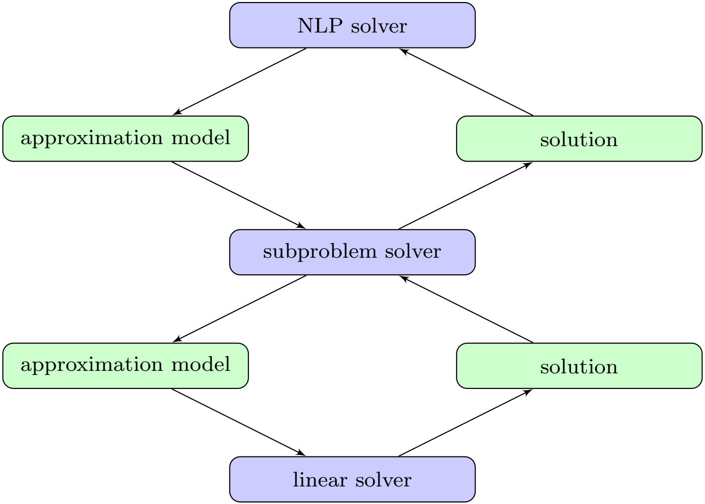
$$\min_{x \in \mathbb{R}^n} \ v(x), \quad \text{where} \quad v(x) = \text{dist}(c(x)|\mathbb{R}^m_-). \tag{FP}$$

Any feasible point for (NLP) is an optimal solution of (FP).

Algorithmic framework: Classic



Algorithmic framework: Detailed



Inefficiencies of traditional approaches

The traditional NLP algorithm classes, i.e.,

- augmented Lagrangian (AL) methods
- sequential quadratic optimization (SQP) methods
- interior-point (IP) methods

may fail or be inefficient when

- exact subproblem solves are expensive
- ...or inexact solves are not computed intelligently
- algorithmic parameters are initialized poorly
- ...or are updated too slowly or inappropriately
- a globalization mechanism inhibits productive early steps
- ...or blocks superlinear local convergence

This is especially important when your subproblems are NLPs!

Contributions

A variety of algorithms and algorithmic tools incorporating/allowing

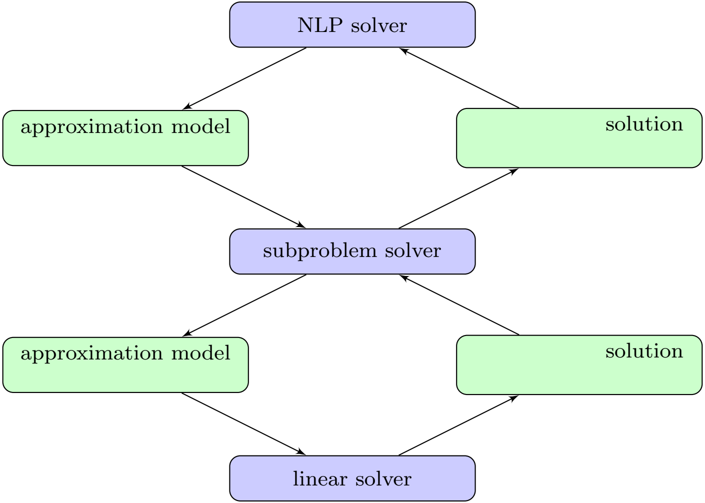
- inexact subproblem solves
- flexible step acceptance strategies
- adaptive parameter updates
- global convergence guarantees
- superlinear local convergence guarantees
- efficient handling of nonconvexity

This talk provides an overview of these tools within

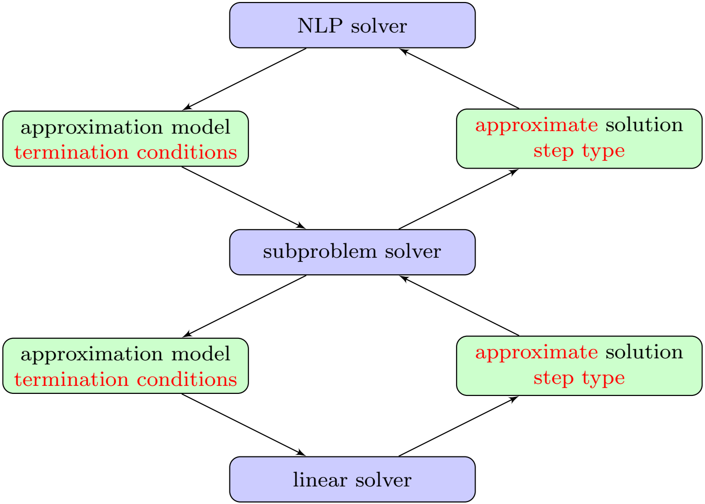
- AL methods with adaptive penalty parameter updates
- SQP methods with inexact subproblem solves
- IP methods with inexact linear system solves
- a penalty-IP method with adaptive parameter updates

and subproblem methods also useful for control, image science, data science, etc.

Algorithmic framework: Detailed



Algorithmic framework: Inexact



Outline

AL methods

A traditional augmented Lagrangian (AL) method for solving

$$\min_{(x,s)\in\mathbb{R}^n\times\mathbb{R}^m_+} f(x) \quad \text{s.t.} \quad c(x)+s=0,$$

observes the following strategy:

- Given $\rho\in\mathbb{R}_+$ and $y\in\mathbb{R}^m$, approximately solve

$$\min_{(x,s)\in\mathbb{R}^n\times\mathbb{R}^m_+} \rho f(x) + (c(x)+s)^T y + \tfrac{1}{2}\|c(x)+s\|_2^2$$

- Update ρ and y to drive (global and local) convergence

Potential inefficiencies:

- Poor initial (ρ, y) may ruin good initial (x, s)
- Slow/poor update for (ρ, y) may lead to poor performance

Ideas: Steering rules and adaptive multiplier updates

"Steering" rules for exact penalty methods: Byrd, Nocedal, Waltz (2008)

- Do not fix ρ during minimization
- Rather, before accepting any step, reduce ρ until the step yields progress in a model of constraint violation proportional to that yielded by a feasibility step
- Cannot be applied directly in an AL method

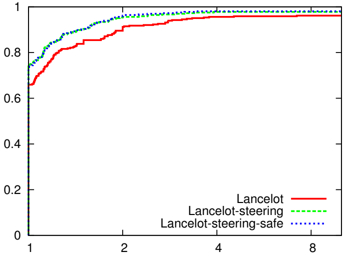
"Steering" rules for AL methods: Curtis, Jiang, Robinson (2014) w/ Gould (2015)

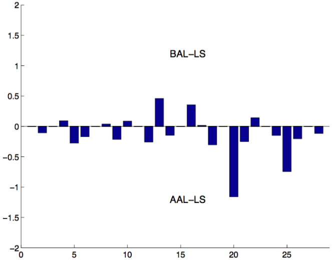
- Modified steering rules that may allow constraint violation increase
- Simultaneously incorporate adaptive multiplier updates
- Gains in performance in trust region and line search contexts
- Implementation in LANCELOT

(Also, "steering" rules in penalty-IP method: Curtis (2012))

Numerical experiments

Implementation in LANCELOT shows that steering (with or without safeguarding) yields improved performance in terms of numbers of iterations for CUTEst set





A MATLAB implementation of an adaptive steering strategy (AAL-LS) outperforms a basic AL method (BAL-LS) in terms of function evaluations on an OPF set

SQP methods

A traditional sequential quadratic optimization (SQP) method:

- Given $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$, solve

$$\min_{d \in \mathbb{R}^n} \; f(x) + g(x)^T d + \tfrac{1}{2} d^T H(x,y) d \quad \text{s.t. } c(x) + J(x)d \leq 0 \tag{QP}$$

- Set $\rho \in \mathbb{R}_+$ to ensure d yields sufficient descent in

$$\phi(x;\rho) = \rho f(x) + v(x)$$

Potential inefficiencies/issues:

- (QP) may be infeasible
- (QP) expensive to solve exactly
- inexact solves might not ensure d yields descent in $\phi(x;\rho)$
- "steering" not viable with inexact solves

Ideas (equalities only): Step decomposition and SMART tests

Step decomposition in trust region (TR) framework: Celis, Dennis, Tapia (1985)

- Normal step toward constraint satisfaction
- Tangential step toward optimality in null space of constraints
- Requires projections during tangential computation

Normal step with TR, but TR-free tangential: Curtis, Nocedal, Wächter (2009)

- Incorporate "SMART" tests: Byrd, Curtis, Nocedal (2008, 2010)
- Normal and tangential steps can be computed approximately
- Consider various types of inexact solutions
- Prescribed inexactness conditions based on penalty function model reduction

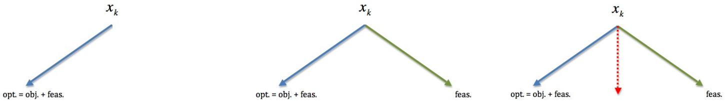
Ideas (inequalities, too): Feasibility and optimality steps w/ scenarios

Inexact SQP method: Curtis, Johnson, Robinson, Wächter (2014)

- Similar to steering, compute approximate(!) feasibility step for reference
- Also given $\rho \in \mathbb{R}_+$, solve

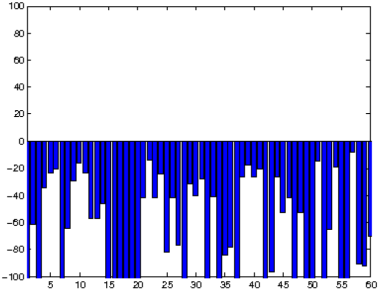
$$\begin{aligned} \min_{d \in \mathbb{R}^n} \ & \rho(f(x) + g(x)^T d) + e^T s + \tfrac{1}{2} d^T H(\rho, x, y) d \\ \text{s.t. } & c(x) + J(x) d \leq s \end{aligned} \qquad \text{(PQP)}$$

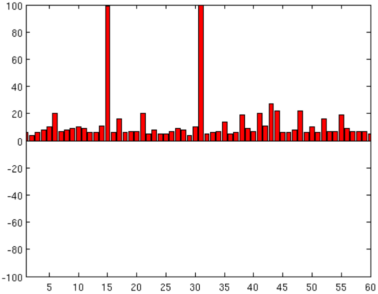
- Consider various types of inexact solutions
  - Approximate Sℓ_1QP step
  - Multiplier-only step
  - Convex combination of feasibility and Sℓ_1QP step



Iteration comparison: AL vs. SQP

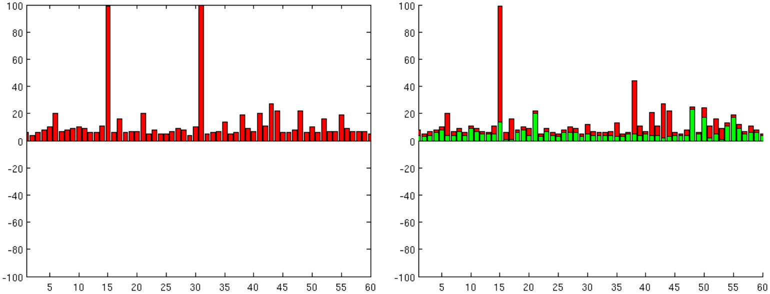
AL (left) with “cheap” iterations vs. SQP (right) with “expensive” iterations





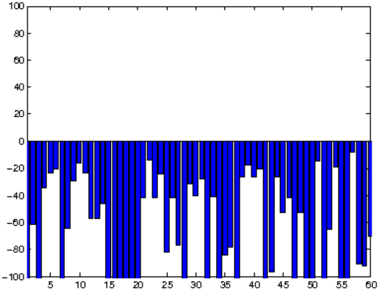
Iteration comparison: SQP vs. iSQP

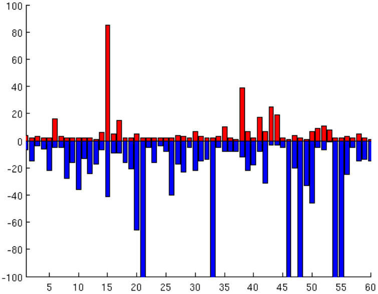
SQP (left) with “expensive” iterations vs. iSQP (right) with “cheaper” iterations



Iteration comparison: AL vs. iSQP

AL (left) with “cheap” iterations vs. iSQP (right) with few “expensive” iterations





IP methods

A traditional interior-point (IP) method for solving

$$\min_{(x,s)\in\mathbb{R}^n\times\mathbb{R}^m_+} f(x) \quad \text{s.t.} \quad c(x)+s=0,$$

observes the following strategy:

- Given $\mu \in \mathbb{R}_+$, approximately solve

$$\min_{(x,s)\in\mathbb{R}^n\times\mathbb{R}^m_+} f(x) - \mu\sum_{i=1}^{m}\ln s^i \quad \text{s.t.} \quad c(x)+s=0$$

- Update μ to drive (global and local) convergence

Potential inefficiences:

- Direct factorizations for Newton's in subproblem can be expensive
- Slack bounds can block long steps ("jamming")
- Slow/poor update for μ may lead to poor performance

Ideas: Inexact IP method

Step decomposition with scaled trust region: Byrd, Hribar, Nocedal (1999)

$$\left\| \begin{bmatrix} d_k^x \\ S_k^{-1} d_k^s \end{bmatrix} \right\|_2 \leq \Delta_k$$

- Allow inexact linear system solves: Curtis, Schenk, Wächter (2010)
- Normal step with TR, but TR-free tangential
- Incorporate (modified) "SMART" tests
- Implemented in IPOPT (optimizer) with inexact, iterative linear system solves by PARDISO (SQMR): Curtis, Huber, Schenk, Wächter (2011)

Implementation details

- Incorporated in IPOPT software package: Wächter, Laird, Biegler
  - interior-point algorithm with inexact step computations
  - flexible penalty function for faster convergence: Curtis, Nocedal (2008)
  - tests on $\sim$ 700 CUTEr problems (almost) on par with original IPOPT
- Linear systems solved with PARDISO: Schenk, Gärtner
  - includes iterative linear system solvers, e.g., SQMR: Freund (1997)
  - incomplete multilevel factorization with inverse-based pivoting
  - stabilized by symmetric-weighted matchings
- Server cooling example coded w/ LIBMESH: Kirk, Peterson, Stogner, Carey

Hyperthermia treatment planning

Let $u_j = a_j e^{i\phi_j}$ and $M_{jk}(x) = \langle E_j(x), E_k(x)\rangle$ where $E_j = \sin(jx_1x_2x_3\pi)$:

$$\min \frac{1}{2}\int_\Omega (y(x) - y_t(x))^2 dx$$
$$\text{s.t.} \begin{cases} -\Delta y(x) - 10(y(x) - 37) = u^* M(x) u & \text{in } \Omega \\ 37.0 \le y(x) \le 37.5 & \text{on } \partial\Omega \\ 42.0 \le y(x) \le 44.0 & \text{in } \Omega_0 \end{cases}$$

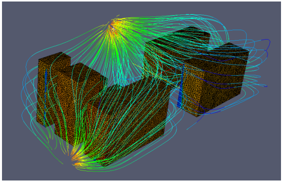
Original IPOPT with $N = 32$ requires 408 seconds per iteration.

N	n	p	q	# iter	CPU sec (per iter)
16	4116	2744	2994	68	22.893 (0.3367)
32	32788	27000	13034	51	3055.9 (59.920)

Server room cooling

Let $\phi(x)$ be the air flow velocity potential:

$$\min \sum c_i v_{AC_i}$$
$$\text{s.t.} \begin{cases} \nabla \phi(x) &= 0 & \text{in } \Omega \\ \partial_n \phi(x) &= 0 & \text{on } \partial\Omega_{wall} \\ \partial_n \phi(x) &= -v_{AC_i} & \text{on } \partial\Omega_{AC_i} \\ \phi(x) &= 0 & \text{in } \Omega_{Exh_j} \\ \|\nabla \phi(x)\|_2^2 &\geq v_{min}^2 & \text{on } \partial\Omega_{hot} \\ v_{AC_i} &\geq 0 & \end{cases}$$



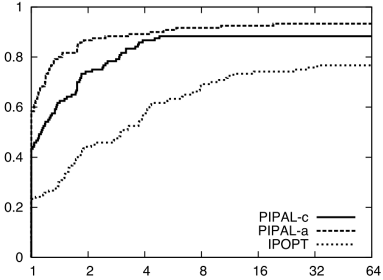
Original IPOPT with $h = 0.05$ requires 2390.09 seconds per iteration.

h	n	p	q	# iter	CPU sec (per iter)
0.10	43816	43759	4793	47	1697.47 (36.1164)
0.05	323191	323134	19128	54	28518.4 (528.119)

Ideas: Penalty-IP method

"Jamming" can be avoided by relaxation via penalty methods

- Two parameters to juggle: ρ and μ
- Simultaneous update motivated by "steering" penalty methods...
- ... and "adaptive barrier" method: Nocedal, Wächter, Waltz (2009)
- ... leading to penalty-interior-point method: Curtis (2012)
- More reliable than IPOPT on degenerate and infeasible problems



Ideas: Trust-funnel IP method

Trust-funnel method for equality constrained problems: Gould, Toint (2010)

- Does not require a penalty function or a filter
- Drives convergence by a monotonically decreasing sequence with

$$v(x_k) \leq v_k^{max} \quad \text{for all} \quad k$$

- Normal and tangential step decomposition, but allows flexible computation that may allow skipping certain subproblems in some iterations
- IP method with scaled trust regions: Curtis, Gould, Robinson, Toint (2015)

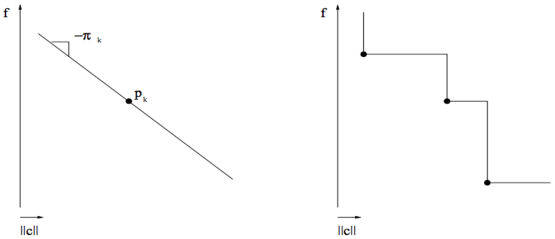
Idea: Flexible penalty function

A traditional penalty/merit function—Han (1977); Powell (1978)—requires

$$f(x_k + \alpha_k d_k) + \pi_k v(x_k + \alpha_k d_k) \ll f(x_k) + \pi_k v(x_k)$$

while a filter—Fletcher, Leyffer (1998)—requires

$$f(x_k + \alpha_k d_k) \ll f_j \quad \text{or} \quad v(x_k + \alpha_k d_k) \ll v_j \quad \text{for all} \quad j \in \mathcal{F}_k$$

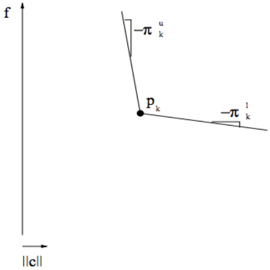


Either can result in "blocking" of certain steps

Idea: Flexible penalty function

A flexible penalty function—Curtis, Nocedal (2008)—requires

$$f(x_k + \alpha_k d_k) + \pi v(x_k + \alpha_k d_k) \ll f(x_k) + \pi v(x_k) \quad \text{for some} \quad \pi \in [\pi_k^l, \pi_k^u]$$



Parameters π_k^l and π_k^u updated separately

Of course, “blocking” may still occur, but (hopefully) less often

Idea: Rapid infeasibility detection

Suppose your algorithm is "driven" by a penalty/merit function such as

$$\phi(x) = \rho f(x) + v(x)$$

and, at an iterate x_k, the following occur:

- x_k is infeasible for (NLP) in that $v(x_k) \gg 0$
- you have computed a feasibility step that does not reduce a model of v sufficiently compared to $v(x_k)$

Then, there is reason to believe that (NLP) is (locally) infeasible, and you may obtain superlinear convergence to an infeasible stationary point by setting

$$\rho \leq \|\text{KKT for (FP)}\|^2$$

Byrd, Curtis, Nocedal (2010); Burke, Curtis, Wang (2014)

Idea: Dynamic Hessian modifications

Inexact SQP and IP methods involve iterative solves of systems of the form

$$\begin{bmatrix} H_k & J_k^T \\ J_k & 0 \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix} = \begin{bmatrix} g_k \\ c_k \end{bmatrix}$$

If H_k is not positive definite in the null space of J_k, then even an exact solution may not be productive in terms of optimization

- Apply a symmetric indefinite iterative solver
- Under certain conditions, modify H_k (say, adding some multiple of a positive definite matrix) and restart the solve (hopefully not entirely from scratch)

Outline

Subproblem solvers for SQP methods

How should QP subproblems be solved within adaptive NLP methods?

- Need scalable step computations
- ... but also the ability to obtain accurate solutions quickly

Traditional approaches have not been able to take SQP methods to large-scale!

- Primal (or dual) active-set methods
- Interior-point methods
- Alternating direction methods (sorry to the ADMM fans out there...)

Iterative reweighting algorithm

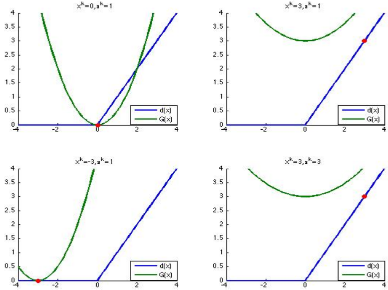
Consider a QP subproblem with $H \succ 0$ of the form

$$\min_{x \in \mathbb{R}^n} \; g^T x + \tfrac{1}{2} x^T H x + \text{dist}(Ax + b \,|\, \mathbb{R}^m_-)$$

Approximate with a smooth quadratic about (x_k, ϵ_k) and solve (as $\{\epsilon_k\} \to 0$)

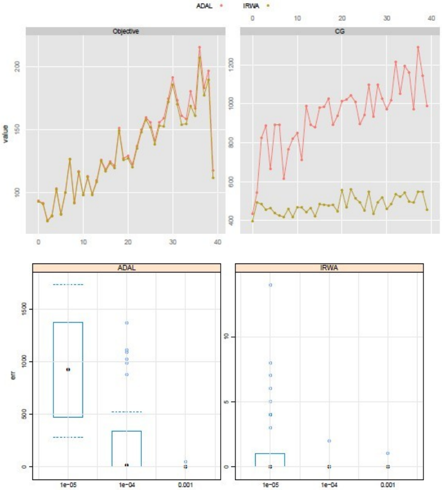
$$\min_{x \in \mathbb{R}^n} \; g^T x + \tfrac{1}{2} x^T H x + \sum_{i=1}^{m} w_i(x_k, \epsilon_k) \| Ax + b - \text{Proj}(Ax_k + b) \|_2^2$$

Convergence/complexity theory, generic dist$(\cdot)$; Burke, Curtis, Wang, Wang (2015)



Numerical experiments

Results for ℓ_1-SVM show improvements over ADMM in CG iterations and sparsity



Primal-dual active-set (PDAS) methods

Consider a QP subproblem with $H \succ 0$ of the form

$$\min_{x \in \mathbb{R}^n} \; g^T x + \tfrac{1}{2} x^T H x \quad \text{s.t.} \quad x \le u$$

Given a partition $(\mathcal{A}, \mathcal{I})$ of the index set of variables, a PDAS method performs:

1. Set $x_{\mathcal{A}} = u_{\mathcal{A}}$ and $y_{\mathcal{I}} = 0$
2. Compute a primal subspace minimizer $x_{\mathcal{I}}$ (via linear system)
3. Set remaining dual variables $y_{\mathcal{A}}$ to satisfy complementarity
4. Update partition based on violated bounds

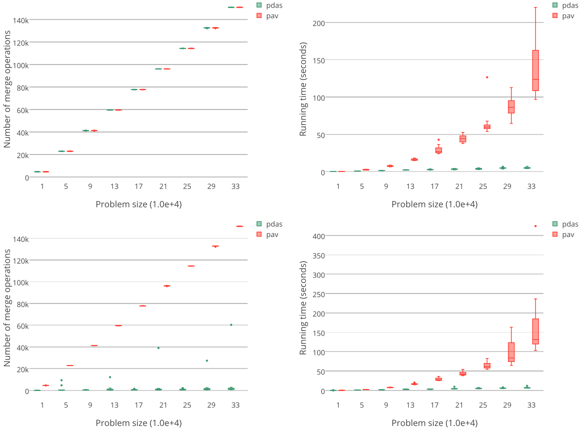
Finite termination for certain H (e.g., M-matrix): Hintermüller, Ito, Kunish (2003)

Algorithmic extensions

We have proposed a variety of algorithmic extensions:

- Globalization strategy for general convex QPs: Curtis, Han, Robinson (2014)
- Inexact subspace minimization techniques (w/ guarantees) for certain H: Curtis, Han (2015)

Adapted for isotonic regression and trend filtering: Curtis, Han (2015)



Outline

Contributions

A variety of algorithms and algorithmic tools incorporating/allowing

- inexact subproblem solves
- flexible step acceptance strategies
- adaptive parameter updates
- global convergence guarantees
- superlinear local convergence guarantees
- efficient handling of nonconvexity

We have outlined these tools within

- an AL method with adaptive penalty parameter updates
- SQP methods with inexact subproblem solves
- an IP method with inexact linear system solves
- a penalty-IP method with dynamic parameter updates

and discussed subproblem methods also useful in their own right

Bibliography I

[1] J. V. Burke, F. E. Curtis, and H. Wang.
A Sequential Quadratic Optimization Algorithm with Rapid Infeasibility Detection.
SIAM Journal on Optimization, 24(2):839–872, 2014.

[2] J. V. Burke, F. E. Curtis, H. Wang, and J. Wang.
Iterative Reweighted Linear Least Squares for Exact Penalty Subproblems on Product Sets.
SIAM Journal on Optimization, 25(1):261–294, 2015.

[3] R. H. Byrd, F. E. Curtis, and J. Nocedal.
An Inexact SQP Method for Equality Constrained Optimization.
SIAM Journal on Optimization, 19(1):351–369, 2008.

[4] R. H. Byrd, F. E. Curtis, and J. Nocedal.
An Inexact Newton Method for Nonconvex Equality Constrained Optimization.
Mathematical Programming, 122(2):273–299, 2010.

[5] R. H. Byrd, F. E. Curtis, and J. Nocedal.
Infeasibility Detection and SQP Methods for Nonlinear Optimization.
SIAM Journal on Optimization, 20(5):2281–2299, 2010.

[6] F. E. Curtis.
A Penalty-Interior-Point Algorithm for Nonlinear Constrained Optimization.
Mathematical Programming Computation, 4(2):181–209, 2012.

[7] F. E. Curtis, N. I. M. Gould, H. Jiang, and D. P. Robinson.
Adaptive Augmented Lagrangian Methods: Algorithms and Practical Numerical Experience.
Technical Report 14T-006, COR@L Laboratory, Department of ISE, Lehigh University, 2014. In second review for *Optimization Methods and Software*.

[8] F. E. Curtis, N. I. M. Gould, D. P. Robinson, and Ph. L. Toint.
An Interior-Point Trust-Funnel Algorithm for Nonlinear Optimization.
Technical Report 13T-010, COR@L Laboratory, Department of ISE, Lehigh University, 2013. In second review for *Mathematical Programming*.

Bibliography II

[9] F. E. Curtis and Z. Han.
Globally Convergent Primal-Dual Active-Set Methods with Inexact Subproblem Solves.
Technical Report 14T-010, COR@L Laboratory, Department of ISE, Lehigh University, 2014. In second review for *SIAM Journal on Optimization*.

[10] F. E. Curtis, J. Huber, O. Schenk, and A. Wächter.
A Note on the Implementation of an Interior-Point Algorithm for Nonlinear Optimization with Inexact Step Computations.
Mathematical Programming, Series B, 136(1):209–227, 2012.

[11] F. E. Curtis, H. Jiang, and D. P. Robinson.
An Adaptive Augmented Lagrangian Method for Large-Scale Constrained Optimization.
Mathematical Programming, DOI: 10.1007/s10107-014-0784-y, 2014.

[12] F. E. Curtis, T. Johnson, D. P. Robinson, and A. Wächter.
An Inexact Sequential Quadratic Optimization Algorithm for Large-Scale Nonlinear Optimization.
SIAM Journal on Optimization, 24(3):1041–1074, 2014.

[13] F. E. Curtis and J. Nocedal.
Flexible Penalty Functions for Nonlinear Constrained Optimization.
IMA Journal of Numerical Analysis, 28(4):749–769, 2008.

[14] F. E. Curtis, J. Nocedal, and A. Wächter.
A Matrix-Free Algorithm for Equality Constrained Optimization Problems with Rank Deficient Jacobians.
SIAM Journal on Optimization, 20(3):1224–1249, 2009.

[15] F. E. Curtis, O. Schenk, and A. Wächter.
An Interior-Point Algorithm for Large-Scale Nonlinear Optimization with Inexact Step Computations.
SIAM Journal on Scientific Computing, 32(6):3447–3475, 2010.