Stanford University, ICME

CME 338 Large-Scale Numerical Optimization

Instructor: Michael Saunders Spring 2019

Notes 12: **MINOS Part 2: LCL Methods**

1 Origins

We now discuss *linearly constrained Lagrangian methods* (LCL methods) for solving the general optimization problem

NCB	$\displaystyle\min_{x\in\mathbb{R}^n}\ \phi(x)$ subject to $c(x)=0,\quad \ell \le x \le u.$

LCL methods seem to have evolved as an alternative to augmented Lagrangian methods, with the early developers being unaware that each method needed features of the other. (As we shall see, some features of the ℓ_1 penalty function are also needed.)

In the augmented Lagrangian approach, the penalty term $\frac{1}{2}\rho_k\|c(x)\|^2$ provides positive curvature to the Lagrangian, allowing a new estimate of the optimal x to be obtained as a *minimizer* of the augmented Lagrangian $L(x, y_k, \rho_k)$ using unconstrained or bound-constrained solvers. The risk is ill-conditioning of the Hessian $\nabla^2_{xx}L(x, y_k, \rho_k)$ if ρ_k is too large or only just large enough to create a local minimizer. A disadvantage is having to optimize within a large space (depending on the number of active bounds).

The methods of Robinson [16] and Robinson and Kreuser [17] made use of *linearized constraints* within the subproblems. They avoided ill-effects from the penalty term by the simple device of not including it(!), and they offered the prospect of optimizing in a smaller space. Suppose (x^*, y^*) solves problem NCB, and suppose the constraints are linearized at x^* to give the linearly constrained problem

LC_*	$\displaystyle\min_{x\in\mathbb{R}^n}\ \phi(x) - {y^*}^T c(x)$ subject to linearized $c(x)=0,\quad \ell \le x \le u.$

A key observation is that x^* solves LC_* as well as NCB. LCL methods build on this observation by defining subproblems like LC_* involving the Lagrangian and the linearization of $c(x)$ at a sequence of points $\{(x_k, y_k)\}$.

The dual vector for LC_* proves to be $\Delta y^* = 0$, implying that the linearized constraints are not constraining the solution when the objective is the "optimal" Lagrangian $L(x, y^*, 0)$. The same is true when the objective is $L(x, y^*, \rho)$ for any $\rho \ge 0$. This gives us scope for satisfying the second-order conditions for optimality.

Early views of LCL methods are well described by Fletcher [3]. The name SLC has been used in the past to indicate *sequential linearly constrained* subproblems that give rise to the next point (x_{k+1}, y_{k+1}). We use the term LCL to highlight the presence of the Lagrangian (or augmented Lagrangian) and the relationship to the BCL approach of LANCELOT.

2 Robinson's method

For a given point (x_k, y_k), define the following linear and nonlinear functions:

Linear approximation to $c(x)$:

$$\bar{c}_k(x) = c(x_k) + J(x_k)(x - x_k)$$

Departure from linearity:

$$d_k(x) = c(x) - \bar{c}_k(x)$$

Modified Lagrangian:

$$M_k(x) = \phi(x) - y_k^T d_k(x)$$

Robinson's method obtains (x_{k+1}, y_{k+1}) by solving the subproblem

$$\text{LC}_k \qquad \begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & M_k(x) \\ \text{subject to} & \bar{c}_k(x) = 0, \quad \ell \le x \le u. \end{array}$$

Conceptually, the objective $M_k(x)$ could be the normal Lagrangian because $d_k(x)$ and $c(x)$ are the same when the linearized constraints are satisfied ($\bar{c}_k(x) = 0$). However, the solution to LC_k would then have to be regarded as $(x_{k+1}, \Delta y_k)$ with $y_{k+1} = y_k + \Delta y_k$.

Under suitable conditions, Robinson [16] proves that the sequence of subproblem solutions $\{(x_k, y_k)\}$ converges *quadratically* to a solution of NCB. Robinson therefore calls it a *Newton-like* method.

A strength of the method is that any convenient solver may be used as a "black box" for the subproblems LC_k. An obvious example is the reduced-gradient method in MINOS. The solver may benefit from working within the reduced subspace defined by the constraint linearization. In addition, rapid convergence is obtainable without the use of second derivatives.

In practice, Robinson's method need not solve LC_k accurately until a solution is approached. It has been observed to succeed remarkably often, in the sense of converging to a local optimum on a wide range of convex and non-convex problems. Note that a good x_0 with $y_0 = 0$ generates a good (x_1, y_1) under favorable conditions.

The choice $y_k = 0$, $\rho_k = 0$ is known to converge for certain *reverse-convex* problems (see references in [15]), but in general, convergence is assured only if (x_0, y_0) is sufficiently close to (x^*, y^*).

3 MINOS

In order to generalize the RG method in MINOS, Robinson's LCL approach was regarded as a promising alternative to GRG, the method implemented in the (excellent) solver CONOPT [1]. As described in Murtagh and Saunders [15], MINOS includes the penalty term of the *augmented* Lagrangian in the subproblem objective in an attempt to improve convergence of the LCL method from arbitrary starting points. A *modified* augmented Lagrangian is used in LC_k:

$$M_k(x) = \phi(x) - y_k^T d_k(x) + \tfrac{1}{2}\rho_k \|d_k(x)\|^2,$$

and again it is equivalent to the normal augmented Lagrangian when $\bar{c}_k(x) = 0$. An important benefit is that if $c(x)$ involves only *some* of the variables nonlinearly, then $M_k(x)$ has the same property, whereas $L(x, y_k, \rho_k) = \phi(x) - y_k^T c(x) + \frac{1}{2}\rho_k \|c(x)\|^2$ is more nonlinear.

3.1 Inexact solution of LC_k

MINOS uses simplex or reduced-gradient iterations to satisfy the linearized constraints for each subproblem. It then limits the number of "minor iterations" performed on LC_k as a heuristic way to avoid excessive optimization within the wrong subspace. MINOS also monitors $\|x_{k+1} - x_k\|$ and $\|y_{k+1} - y_k\|$ and if they seem large, the step toward (x_{k+1}, y_{k+1}) is heuristically shortened.

By analogy with other augmented Lagrangian methods, it would probably help to check $\|c(x)\|$ when each subproblem is terminated. If $\|c(x)\|$ has increased substantially, ρ_k could be increased.

3.2 Reducing ρ_k

In practice it is readily observed that if the LCL subproblems converge, they generally do so more quickly if ρ_k is not large. By default, ρ_0 is initially moderate ($100/m_1$ for problems with m_1 nonlinear constraints). When $\|c(x)\|$ and $\|y_{k+1} - y_k\|$ become reasonably small, MINOS reduces ρ_k in stages. It was originally thought that $\rho_k = 0$ was needed to retain the quadratic convergence of Robinson's method, but a small positive value $\rho_k = \bar{\rho}$ is currently retained after several reductions. As observed by Friedlander [4], Robinson's convergence analysis applies to problem NCB with objective $\bar{\phi}(x) \equiv \phi(x) + \frac{1}{2}\bar{\rho}\|c(x)\|^2$ (and $c(x) = 0$ at a solution).

3.3 Infeasible subproblems

Perhaps the greatest source of difficulty with Robinson's method (including its implementation in MINOS) is the occurrence of infeasible linearizations. Heuristically, MINOS relaxes the linearized constraints in stages. If the constraints remain infeasible after five consecutive relaxations, the original problem is decleared infeasible. Otherwise, the relaxed LC_k is optimized as usual. Future subproblems are not relaxed unless they are also infeasible.

The relaxation process is successful sometimes, but in practice it is safer for users to add their own "elastic slacks" on troublesome nonlinear constraints, with a linear cost (penalty) on the elastic slacks. The subproblems are then automatically feasible in a relaxed form where necessary, and the relaxation is kept to a minimum by the penalties.

A similar process is automated within SNOPT [6]. All nonlinear constraints are made equally elastic if LC_k is infeasible, or if $\|y_k\|$ grows large (a sign of impending infeasibility).

Elastic constraints are an even more integral part of the *stabilized LCL method* developed in Friedlander's thesis research [4]. Indeed, all of the heuristic maneuvers listed above are dealt with methodically by Friedlander's sLCL algorithm. The resulting insights came 20 years after the first LCL version of MINOS, and 30 years after Robinson's tantalizing analysis of the local convergence properties of his Newton-like (but not exactly Newton) process.

4 A stabilized LCL method

In Friedlander and Saunders [5], the elastic LC subproblem is written in terms of the normal augmented Lagrangian

$$L_k(x) \equiv L(x, y_k, \rho_k) = \phi(x) - y_k^T c(x) + \tfrac{1}{2}\rho_k\|c(x)\|^2$$

as follows:

$$\boxed{(\mathrm{ELC}_k) \qquad \begin{array}{ll} \underset{x,v,w}{\text{minimize}} & L_k(x) + \sigma_k e^T(v+w) \\ \text{subject to} & \bar{c}_k(x) + v - w = 0, \quad \ell \le x \le u, \quad v, w \ge 0, \end{array}}$$

where e is a vector of 1s and $\sigma_k > 0$. This subproblem is *always feasible.* Its solution is of the form $(x_k^*, \Delta y_k^*, v_k^*, w_k^*)$, with σ_k having the effect of enforcing $\|\Delta y_k^*\|_\infty \le \sigma_k$.

The subproblem is also stated in terms of the ℓ_1 penalty function:

$$\boxed{(\mathrm{ELC}'_k) \qquad \begin{array}{ll} \underset{x}{\text{minimize}} & L_k(x) + \sigma_k\|\bar{c}_k(x)\|_1 \\ \text{subject to} & \ell \le x \le u. \end{array}}$$

This form reveals a strong connection with the BCL approach. Indeed, the strategies for setting ρ_k and for controlling the accuracy of the subproblem solutions closely follow the BCL algorithm in LANCELOT. The primary innovation is the ℓ_1 penalty term. Far from a solution, this term allows the method to deviate from the constraint linearizations. Near a solution, it keeps the iterates close to the linearizations. For values of σ_k above a certain threshold, the linearized constraints are satisfied exactly (assuming the original constraints are feasible), thus permitting the rapid convergence of Robinson's method.

Experience with MINOS and SNOPT suggests that ρ_k should be *reduced* a finite number of times as convergence takes place, to make the subproblems easier to solve.

Returning to (ELC_k), we may subtract the elastic linearized constraints from both occurrences of $c(x)$ in $L_k(x)$. We now have the following functions:

Linear approximation to $c(x)$:

$$\bar{c}_k(x) = c(x_k) + J(x_k)(x - x_k)$$

Elastic departure from linearity:

$$d_k(x,v,w) = c(x) - \bar{c}_k(x) - v + w$$

Modified augmented Lagrangian:

$$\begin{aligned}
M_k(x,v,w) &= \phi(x) - y_k^T d_k(x,v,w) + \tfrac{1}{2}\rho_k \|d_k(x,v,w)\|^2 \\
\nabla M_k(x) &= g_0(x) - \big(J(x) - J_k\big)^T \widetilde{y}_k(x,v,w) \\
\nabla^2 M_k(x) &= H_0(x) - \textstyle\sum_i (\widetilde{y}_k(x))_i H_i(x) + \rho_k \big(J(x) - J_k\big)^T \big(J(x) - J_k\big) \\
\widetilde{y}_k(x,v,w) &\equiv y_k - \rho_k d_k(x,v,w)
\end{aligned}$$

With these definitions, a sequence $\{(x_k, y_k)\}$ may be defined from increasingly accurate solutions (x_k^*, y_k^*) to elastic linearized subproblems of the form

$$\boxed{\begin{array}{lll}
(\mathrm{ELC}_k'') & \underset{x,\,v,\,w}{\text{minimize}} & M_k(x,v,w) + \sigma_k e^T(v+w) \\
& \text{subject to} & \bar{c}_k(x) + v - w = 0, \quad \ell \le x \le u, \quad v, w \ge 0.
\end{array}}$$

4.1 The sLCL algorithm

The stabilized LCL algorithm [4, 5] is virtually the same as the BCL algorithm except for the elastic subproblem (ELC_k'') and manipulation of the associated parameter σ_k. The subproblem dual variables y_k^* and z_k^* give y_{k+1} and z_{k+1} directly.

Algorithm 1: sLCL (Stabilized LCL Method).

Input: x_0, y_0, z_0
Output: x^*, y^*, z^*

Set $\bar{\sigma} \gg 1$ and scale factors $\tau_\rho, \tau_\sigma > 1$.
Set penalty parameters $\rho_1 > 0$ and $\sigma_1 \in [1, \bar{\sigma}]$.
Set positive convergence tolerances $\omega_*, \eta_* \ll 1$ and infeasibility tolerance $\eta_1 > \eta_*$.
Set constants $\alpha, \beta > 0$ with $\alpha < 1$.
$k \leftarrow 0$, converged $\leftarrow$ false

repeat
  $k \leftarrow k + 1$
  Choose optimality tolerance $\omega_k > 0$ such that $\lim_{k\to\infty} \omega_k \le \omega_*$.
  Find $(x_k^*, v_k^*, w_k^*, y_k^*, z_k^*)$ that solves (ELC_k'') to within tolerance ω_k.
  if $\|c(x_k^*)\| \le \max(\eta_*, \eta_k)$ **then**
    $x_k \leftarrow x_k^*,\ \ y_k \leftarrow y_k^*,\ \ z_k \leftarrow z_k^*$ [update solution estimates]
1   **if** (x_k, y_k, z_k) solves NCB **then** converged $\leftarrow$ true
    $\Delta y_k^* \leftarrow y_k^* - y_k + \rho_k c_k(x_k^*)$
2    $\rho_{k+1} \leftarrow \rho_k$ [keep ρ_k]
3    $\sigma_{k+1} \leftarrow \frac{1}{1+\rho_k} \min(1 + \|\Delta y_k^*\|_\infty, \bar{\sigma})$ [reset σ_k]
    $\eta_{k+1} \leftarrow \eta_k / (1 + \rho_{k+1}^\beta)$ [decrease η_k]
  else
    $\rho_{k+1} \leftarrow \tau_\rho \rho_k$ [increase ρ_k]
    $\sigma_{k+1} \leftarrow \sigma_k / \tau_\sigma$ [decrease σ_k]
    $\eta_{k+1} \leftarrow \eta_0 / (1 + \rho_{k+1}^\alpha)$ [may increase or decrease η_k]
  end
until converged
$x^* \leftarrow x_k,\ \ y^* \leftarrow y_k,\ \ z^* \leftarrow z_k$

Key features are that the subproblems are solved inexactly (they are always feasible) and the penalty parameter ρ_k is increased only finitely often. The quantity $\|\Delta y_k^*\|$ is bounded by $\sigma_k + \omega_k$ and hence is uniformly bounded for large enough k. Under certain conditions, eventually $v_k = w_k = 0$, the ρ_k's remain constant (and could be reduced to improve efficiency), the iterates converge quadratically as in Robinson's method, and the algorithm terminates in a finite number of (major) iterations.

4.2 Infeasible problems

As given above, the sLCL algorithm might not terminate if problem NCB has no solution. An additional test is needed to force a "Problem is infeasible" exit if ρ_k is above a certain threshold and $\|c(x_k^*)\|$ is consistently not decreasing. In [4, 5] it is shown that on infeasible problems, the sLCL algorithm converges to a local minimizer or stationary point for the function $\|c(x)\|_2^2$. In contrast, SNOPT minimizes $\|c(x)\|_1$.

4.3 Summary

The Conclusions in [5] include the following notes:

The stabilized LCL method is a generalization of augmented Lagrangian methods and it shares the strengths of its predecessors: it is globally convergent (the BCL advantage) and it has quadratic local convergence (the LCL advantage). The ℓ_1 penalty function on the linearized constraints brings the two together. Because the method operates in a reduced space (like all LCL methods), it is less sensitive than BCL methods to the choice of each penalty parameter ρ_k. $\cdots$

The sLCL method is largely independent of the method by which its subproblems are solved. An LC solver using second derivatives is likely to require fewer iterations (and hence less computational work) for the solution of each of the subproblem. We would expect the number of required major iterations to remain constant if each subproblem solution is computed to within the prescribed tolerance ω_k. However, we would expect to *reduce* the number of required major iterations if a MINOS-like strategy is used to terminate the subproblems (limiting the number of minor iterations). Over the same number of iterations, a subproblem solver using second derivatives may make more progress toward a solution than a first-derivative solver.

4.4 Looking ahead

A Fortran 90 implementation of the sLCL algorithm based on subproblem (ELC$''_k$) has been developed by Friedlander. Called KNOSSOS (home of King Minos), it allows for a choice of LC subproblem solvers: the reduced-gradient method of MINOS, the SQP method of SNOPT, and perhaps future solvers that make use of second derivatives. KNOSSOS could be reimplemented in the way that AMPL was use to implement NCL [12, 13, 14].

For 40 years, MINOS has been an important nonlinear solver for GAMS and AMPL, even though it may fail on problems with highly nonlinear constraints, especially if they are infeasible. (CONOPT has been especially important for such cases.) SNOPT has demonstrated great robustness by solving 90% of the 1000 CUTEr test problems [7], using remarkably few function and gradient evaluations [6]. A concern is that SNOPT sometimes needs rather many major iterations (and hence linear algebra) compared to MINOS. KNOSSOS/MINOS and KNOSSOS/SNOPT promise the best of both worlds.

Meanwhile, nonlinear interior methods such as IPOPT, KNITRO, and LOQO [9, 10, 11] show considerable promise for applications where second derivatives are available. Comparisons are possible with the aid of CUTEst, the Constrained and Unconstrained Testing Environment with safe threads [8, 2]. CUTEst contains 200 additional test problems, with interfaces to IPOPT, KNITRO, SNOPT, filterSQP, and many other solvers: ALGENCAN, BOBYQA, Direct Search, filterSD, NEWUOA, NLPQLP, NOMAD, PENNLP, QL, SPG, SQIC.

References

[1] ARKI Consulting & Development A/S. `http://www.conopt.com`.

[2] CUTEst on GitHub. `https://github.com/ralna/CUTEst`.

[3] R. Fletcher. Methods related to Lagrangian functions. In P. E. Gill and W. Murray, editors, *Numerical Methods for Constrained Optimization*, pages 219–239. Academic Press, London, 1974.

[4] M. P. Friedlander. *A Globally Convergent Linearly Constrained Lagrangian Method for Nonlinear Optimization*. PhD thesis, Dept of Management Science and Engineering, Stanford University, Aug 2002.

[5] M. P. Friedlander and M. A. Saunders. A globally convergent linearly constrained Lagrangian method for nonlinear optimization. *SIAM J. Optim.*, 15(3):863–897, 2005.

[6] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005. SIGEST article.

[7] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEr and SifDec: A constrained and unconstrained testing environment, revisited. *ACM Trans. Math. Software*, 29(4):373–394, 2003.

[8] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEst: a Constrained and Unconstrained Testing Environment with safe threads. *Comput. Optim. Appl.*, 60:545–557, 2015.

[9] IPOPT open source NLP solver. `https://projects.coin-or.org/Ipopt`.

[10] KNITRO optimization software. `https://www.artelys.com/tools/knitro`.

[11] LOQO optimization software. `http://orfe.princeton.edu/~loqo`.

[12] D. Ma, K. L. Judd, D. Orban, and M. A. Saunders. Algorithm NCL for constrained optimization. `http://stanford.edu/group/SOL/talks.html`, 2018.

[13] D. Ma, K. L. Judd, D. Orban, and M. A. Saunders. Stabilized optimization via an NCL algorithm. `http://stanford.edu/group/SOL/multiscale/models/NCL/`, 2018.

[14] D. Ma, K. L. Judd, D. Orban, and M. A. Saunders. Stabilized optimization via an NCL algorithm. In M. Al-Baali, Lucio Grandinetti, and Anton Purnama, editors, *Numerical Analysis and Optimization, NAO-IV, Muscat, Oman, January 2017*, volume 235 of *Springer Proceedings in Mathematics and Statistics*, pages 173–191. Springer International Publishing Switzerland, 2018.

[15] B. A. Murtagh and M. A. Saunders. A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints. *Math. Program. Study*, 16:84–117, 1982.

[16] S. M. Robinson. A quadratically-convergent algorithm for general nonlinear programming problems. *Math. Program.*, 3:145–156, 1972.

[17] J. B. Rosen and J. Kreuser. A gradient projection algorithm for nonlinear constraints. In F. A. Lootsma, editor, *Numerical Methods for Nonlinear Optimization*, pages 297–300. Academic Press, London and New York, 1972.