

Notes 9: **Augmented Lagrangian Methods**

1 Origins

Large-scale solvers such as CONOPT [6, 1], LANCELOT [3, 4, 9], MINOS [10, 11], and SNOPT [7, 8] are designed to solve optimization problems in the following fairly general form:

NCO	$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{minimize}} && \phi(x) \\ & \text{subject to} && \ell \leq \begin{pmatrix} x \\ Ax \\ c(x) \end{pmatrix} \leq u, \end{aligned}$
-----	---

where A is a sparse matrix, $c(x)$ is a vector of smooth nonlinear functions, and the bounds ℓ and u are likely to exist as shown. (As always, some bounds could be infinite and some could imply equality constraints: $\ell_j = u_j$.)

In the early days of optimization, it was a challenge to optimize a nonlinear function $\phi(x)$ with *no constraints*. Bounds on the variables ($\ell_j \leq x_j \leq u_j$) could be accommodated with reasonable ease by active-set strategies, but more general constraints tended to be imposed via *penalty terms* and *barrier terms* in the objective.

In many cases it is general enough to deal with *nonlinear equality constraints* and simple bounds:

NCB	$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{minimize}} && \phi(x) \\ & \text{subject to} && c(x) = 0, \quad \ell \leq x \leq u, \end{aligned}$
-----	---

where $c(x) \in \mathbb{R}^m$ and x includes slack variables if necessary to deal with inequalities.

Some applications require all iterates to be *feasible*. FSQP is one solver that achieves this. The Generalized Reduced Gradient (GRG) approach exemplified by CONOPT is slightly different. Assuming the current point is feasible, it generates the next search direction by assuming $c(x)$ is linear. It may evaluate $c(x + \alpha\Delta x)$ at infeasible points, but it restores feasibility at each stage before “moving on”.

Otherwise, most methods are content to satisfy $c(x) = 0$ only in the limit. The main approaches of interest are *penalty methods*, *augmented Lagrangian methods* (which solve sequences of optimization subproblems) and *nonlinear interior methods* (which solve sequences of nonlinear equations).

1.1 The Lagrangian

A fundamental function associated with problem NCB (with or without the bound constraints) is the *Lagrangian*:

$$L(x, y) = \phi(x) - y^T c(x).$$

For a given vector y , its derivatives $\nabla_x L$ and $\nabla_{xx}^2 L$ are

$$g_L(x, y, \rho) = g_0(x) - J(x)^T y, \quad (1)$$

$$H_L(x, y, \rho) = H_0(x) - \sum y_i H_i(x). \quad (2)$$

where $g_0(x) = \nabla \phi(x)$ is the objective gradient, $H_0(x) = \nabla^2 \phi(x)$ is the objective Hessian, $J(x)$ is the matrix of constraint gradients (the Jacobian), and $H_i(x)$ is Hessian for the i th constraint function $c_i(x)$. In general, y will be an approximation to the dual variables associated with the constraints $c(x) = 0$.

2 Penalty Methods

For this section we assume there are only equality constraints $c_i(x) = 0$, $i = 1 : m$:

NEC	$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{minimize}} && \phi(x) \\ & \text{subject to} && c(x) = 0. \end{aligned}$
-----	---

A pair (x, y) satisfies the *first-order KKT conditions* for NEC if

$$c(x) = 0, \quad (3)$$

$$J(x)^T y = g_0(x). \quad (4)$$

In real life we are always finding a balance between what is desirable (our objective function) and what is legally achievable (the constraints that prevent infinite profit!). This *multiple objective* point of view suggests solving the unconstrained problem

PP(ρ)	$\underset{x}{\text{minimize}} \quad P(x, \rho) = \phi(x) + \frac{1}{2} \rho \ c(x)\ ^2$
--------------	--

for some penalty parameter $\rho > 0$. $P(x, \rho)$ is the *quadratic penalty function*.

We envisage a trajectory of points x_ρ that solve PP(ρ) for an increasing sequence of ρ values. We must let ρ become large to achieve near feasibility, but at least the penalty function is *smooth*. We may therefore apply Newton or quasi-Newton methods for unconstrained optimization. The derivatives of the penalty function are

$$\begin{aligned} g(x, \rho) &\equiv \nabla P(x, \rho) = g_L(x, y, \rho), \\ H(x, \rho) &\equiv \nabla^2 P(x, \rho) = H_L(x, y, \rho) + \rho J(x)^T J(x), \end{aligned}$$

where g_L and H_L are the Lagrangian derivatives (1)–(2) with $y = -\rho c(x)$ in this case. Note that $g(x, \rho) = 0$ at x_ρ (an unconstrained minimizer of the penalty function). Defining $y_\rho = -\rho c(x_\rho)$, we see that (x_ρ, y_ρ) is the *exact solution* of a perturbed form of problem NEC:

NEC $_\rho$	$\begin{aligned} & \underset{x}{\text{minimize}} && \phi(x) \\ & \text{subject to} && c(x) = c(x_\rho). \end{aligned}$
-------------	--

Also, if the Jacobian $J(x^*)$ has full row rank and x^* is a unique local minimizer for NEC (i.e., the reduced Hessian for NEC is positive definite), we can show that the full Hessian $H(x, \rho)$ is positive definite at (x_ρ, y_ρ) for sufficiently large ρ . Thus, the penalty function is *convex* for large ρ , and the minimizer x_ρ exists.

Newton's method for obtaining a search direction Δx wants to solve the system

$$(H_L + \rho J^T J) \Delta x = -g(x, \rho), \quad (5)$$

which is ill-conditioned for large ρ (if $m \neq n$). This is one reason why the early unconstrained optimizers proved unsuccessful with quadratic penalty functions. It was some time before a cure for the ill-conditioning was recognized, but indeed there is one. Defining $y = -\rho c$ and $\Delta y = -\rho J \Delta x$ at the current x , we see that the Newton system is equivalent to

$$\begin{pmatrix} H_L & J^T \\ J & -\frac{1}{\rho} I \end{pmatrix} \begin{pmatrix} \Delta x \\ -\Delta y \end{pmatrix} = - \begin{pmatrix} g_0 - J^T y \\ 0 \end{pmatrix}, \quad (6)$$

which contains no large numbers and may be preferable for sparsity reasons anyway.

Unfortunately, this doesn't save the quadratic penalty method. Δx moves too much in the null space of the current J . Newton's method is too slow with large ρ .

3 Equality Constraints

We continue to study problems with nonlinear equality constraints, bearing in mind the difficulties encountered with the quadratic penalty method when ρ becomes very large. Suppose (x^*, y^*) is a local minimizer for the problem NEC above, and assume that the Jacobian $J(x) = \nabla c(x)$ has full row rank at $x = x^*$. The first-order optimality conditions that (x^*, y^*) must satisfy are

$$c(x) = 0, \quad (7)$$

$$g_0(x) - J(x)^T y = 0. \quad (8)$$

The Lagrangian associated with NEC is

$$L(x, y) = \phi(x) - y^T c(x).$$

We see that the Lagrangian gradient $\nabla_x L$ must be zero at (x^*, y^*) . The required solution is a *stationary point* of the Lagrangian. However, in general we cannot find x^* by minimizing $L(x, y)$ as a function of x , even if we set $y = y^*$. The required x^* may be a saddle point or even a *maximizer* of the Lagrangian.

The second-order optimality condition for x^* to be an *isolated local minimizer* is that the Lagrangian Hessian $H_L \equiv \nabla_{xx}^2 L$ should be positive definite within the null space of $J(x^*)$. That is, the Hessian should satisfy $z^T H_L z > 0$ for all nonzero vectors z satisfying $J(x^*)z = 0$.

The following result on quadratic forms is relevant.

Theorem 1 (Debrey [5]) *Let H be an $n \times n$ symmetric matrix and J an $m \times n$ matrix with $m \leq n$. If $z^T H z > 0$ for every nonzero z satisfying $Jz = 0$, then for all ρ sufficiently large, $H + \rho J^T J$ is positive definite.*

This suggests that we should add to the Lagrangian a term whose Hessian is $\rho J(x)^T J(x)$. We already know such a function: it appears in the quadratic penalty method.

3.1 The Augmented Lagrangian

The *augmented Lagrangian* associated with NEC is

$$L(x, y, \rho) = \phi(x) - y^T c(x) + \frac{1}{2} \rho \|c(x)\|^2.$$

It may be thought of as a modification to the Lagrangian, *or* as a shifted quadratic penalty function. For a given y and ρ , its derivatives $\nabla_x L$ and $\nabla_{xx}^2 L$ are

$$g_L(x, y, \rho) = g_0(x) - J(x)^T \hat{y}, \quad (9)$$

$$H_L(x, y, \rho) = H_0(x) - \sum \hat{y}_i H_i(x) + \rho J(x)^T J(x), \quad (10)$$

$$\hat{y} \equiv y - \rho c(x). \quad (11)$$

The *augmented Lagrangian method* for solving NEC proceeds by choosing y and ρ judiciously and then minimizing $L(x, y, \rho)$ as a function of x . The resulting x is used to choose a new y and ρ , and the process repeats. The auxiliary vector \hat{y} simplifies the above notation and proves to be useful in its own right.

Note that if ρ is reasonably large, minimizing L will tend to make $\|c(x)\|$ small (as for the penalty method), *even if y is somewhat arbitrary*. Also, H_L will tend to have positive curvature in the right space and a minimizer is likely to exist.

On the other hand, if y is close to y^* , since minimizing L makes $\|g_L\|$ small, we see that $(x, y) \approx (x, y^*)$ almost satisfies (8). If it also happens that $\|c(x)\|$ is small (because ρ is large enough), (x, y) will almost satisfy (7) as well.

The strategy is to check that $\|c(x)\|$ is suitably small after each (approximate) minimization of L . If so, y is updated to $\hat{y} = y - \rho c(x)$. If not, ρ is increased and y remains the same. Under favorable conditions, $(x, y) \rightarrow (x^*, y^*)$ before ρ becomes too large.

4 LANCELOT

We now return the general optimization problem

NCB	$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{minimize}} && \phi(x) \\ & \text{subject to} && c(x) = 0, \quad \ell \leq x \leq u, \end{aligned}$
-----	---

where $c(x) = 0$ includes linear and nonlinear constraints, and x includes slack variables where necessary to deal with inequalities. The large-scale solver LANCELOT

[3, 4, 9] treats NCB by applying an augmented Lagrangian method to a sequence of *bound-constrained subproblems* of the form

$$\begin{array}{ll} \text{(BC}_k\text{)} & \underset{x}{\text{minimize}} \quad L(x, y_k, \rho_k) = \phi(x) - y_k^T c(x) + \frac{1}{2} \rho_k \|c(x)\|^2 \\ & \text{subject to} \quad \ell \leq x \leq u. \end{array}$$

A large-scale trust-region method is applied to each BCL subproblem. It takes care of the bound constraints directly. (They are not part of the augmented Lagrangian.)

We call the LANCELOT approach a *bound-constrained Lagrangian method*, in anticipation of other methods that minimize the augmented Lagrangian subject to additional constraints (which are likely to be linear).

4.1 The BCL Algorithm

Each subproblem (BC_k) is solved with a specified optimality tolerance ω_k , generating an iterate x_k^* and the associated Lagrangian gradient $z_k^* \equiv \nabla L(x_k^*, y_k, \rho_k)$. If $\|c(x_k^*)\|$ is sufficiently small, the iteration is regarded as “successful” and an update to y_k is computed from x_k^* . Otherwise, y_k is not altered but ρ_k is increased.

Key properties are that the subproblems are solved inexactly, the penalty parameter is increased only finitely often, and the multiplier estimates y_k need not be assumed bounded. Under certain conditions, all iterations are eventually successful, the ρ_k 's remain constant, the iterates converge superlinearly, and the algorithm terminates in a finite number of iterations.

4.2 Solving the BCL Subproblems

LANCELOT contains a solver SBMIN for minimizing a nonlinear objective function subject to bounds. SBMIN is used to obtain an approximate solution for each BCL subproblem. It employs an elaborate active-set strategy (involving generalized Cauchy points and a trust-region constraint) to determine a set of active bounds. It then applies a modified Newton or quasi-Newton method to optimize the augmented Lagrangian objective $L(x, y, \rho)$ with respect to the free variables.

In MATLAB notation, most of the sparse-matrix computation is performed in solving linear systems of the form

$$H_L(f, f) \Delta x_f = -g_L(f),$$

where f is an index set denoting rows and columns corresponding to free variables. (Reduced Hessians $Z^T H Z$ and reduced gradients $Z^T g$ are easy to form when the constraints are simple bounds!)

The sparse-matrix package MA27 is used to factorize $H(f, f)$ or $H(f, f) + E$, where E is a diagonal modification to make the system positive definite. As f changes, Schur-complement updates are made in order to re-use the factors.

For very large problems, the symmetric conjugate-gradient (CG) method is used to compute Δx_f , with assistance from a range of preconditioning options. If

Algorithm 1: BCL.

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

Set penalty parameter $\rho_0 > 0$ and scale factor $\tau > 1$.
Set positive convergence tolerances $\eta_*, \omega_* \ll 1$ and infeasibility tolerance $\eta_0 > \eta_*$.
Set constants $\alpha, \beta > 0$ with $\alpha < 1$.
 $k \leftarrow 0$
converged \leftarrow false

repeat

 Choose optimality tolerance $\omega_k > 0$ such that $\lim_{k \rightarrow \infty} \omega_k \leq \omega_*$.
 Find a point (x_k^*, z_k^*) that solves (BC_k) within tolerance ω_k .

if $\|c(x_k^*)\| \leq \max(\eta_*, \eta_k)$ **then**

$y_k^* \leftarrow y_k - \rho_k c(x_k^*)$
 $x_{k+1} \leftarrow x_k^*, y_{k+1} \leftarrow y_k^*, z_{k+1} \leftarrow z_k^*$ [update solution estimates]
 if $(x_{k+1}, y_{k+1}, z_{k+1})$ solves NCB **then** converged \leftarrow true
 $\rho_{k+1} \leftarrow \rho_k$ [keep ρ_k]
 $\eta_{k+1} \leftarrow \eta_k / (1 + \rho_{k+1}^\beta)$ [decrease η_k]

else

$x_{k+1} \leftarrow x_k, y_{k+1} \leftarrow y_k, z_{k+1} \leftarrow z_k$ [keep solution estimates]
 $\rho_{k+1} \leftarrow \tau \rho_k$ [increase ρ_k]
 $\eta_{k+1} \leftarrow \eta_0 / (1 + \rho_{k+1}^\alpha)$ [may increase or decrease η_k]

end
 $k \leftarrow k + 1$

until converged

$x^* \leftarrow x_k, y^* \leftarrow y_k, z^* \leftarrow z_k$

$H_L(f, f)$ is not positive definite, the CG method may terminate with a direction of infinite descent or a direction of negative curvature.

A feature of LANCELOT and SBMIN is their provision for nonlinear functions that are *group partially separable*. This facilitates the formation of g_L and H_L . It is also useful for computing the matrix-vector products $H_L(f, f)v$ required by CG.

5 Partial Augmented Lagrangians

Subproblem BCL above treats the bound constraints directly. Similar theory applies if other constraints are imposed directly rather than via the augmented Lagrangian.

For example, Conn et al. [2] show how to treat general *linear* constraints directly as well as bounds. Probably this was inspired by the strangely poor performance of LANCELOT on linear programs. (However, the approach has not been implemented.)

An important application of this approach is to *nonlinear network problems with side constraints*. Network constraints are very well suited to the reduced-gradient method because the null-space operator Z is extremely sparse and can be formed explicitly. General linear and nonlinear constraints are best included within the augmented Lagrangian objective.

References

- [1] ARKI Consulting & Development A/S. <http://www.conopt.com>.
- [2] A. R. Conn, N. I. M. Gould, A. Sartenaer, and Ph. L. Toint. Convergence properties of an augmented Lagrangian algorithm for optimization with a combination of general equality and linear constraints. *SIAM J. Optim.*, 6:674–703, 1996.
- [3] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM J. Numer. Anal.*, 28:545–572, 1991.
- [4] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *LANCELOT: A Fortran Package for Large-scale Nonlinear Optimization (Release A)*. Lecture Notes in Computation Mathematics 17. Springer Verlag, Berlin, Heidelberg, New York, London, Paris and Tokyo, 1992.
- [5] G. Debreu. Definite and semidefinite quadratic forms. *Econometrica*, 20:295–300, 1952.
- [6] A. Drud. CONOPT: A GRG code for large sparse dynamic nonlinear optimization problems. *Math. Program.*, 31:153–191, 1985.
- [7] 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.
- [8] P. E. Gill, W. Murray, and M. A. Saunders. User’s guide for SNOPT 7: Software for large-scale nonlinear programming. <http://www.scicomp.ucsd.edu/~peg> (see Software), 2006.
- [9] LANCELOT optimization software. <http://www.numerical.rl.ac.uk/lancelot/blurb.html>.
- [10] B. A. Murtagh and M. A. Saunders. Large-scale linearly constrained optimization. *Math. Program.*, 14:41–72, 1978.
- [11] 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.