Practical Optimization Algorithms and Applications

Chapter XII: Penalty and Augmented Lagrangian Methods

Lingfeng NIU

Research Center on Fictitious Economy & Data Science,
University of Chinese Academy of Sciences

niulf@ucas.ac.cn

Outline

Outline

Motivation

One fundamental approach to constrained optimization is to replace the original problem by a penalty function that consists of

- the original objective of the constrained optimization problem, plus
- one additional term for each constraint, which is positive when the current point x violates that constraint and zero otherwise.

Most approaches define a *sequence* of such penalty functions, in which the penalty terms for the constraint violations are multiplied by some positive coefficient. By making this coefficient larger and larger, we penalize constraint violations more and more severely, thereby forcing the minimizer of the penalty function closer and closer to the feasible region for the constrained problem.

Motivation

The simplest penalty function of this type is the quadratic penalty function, in which the penalty terms are the squares of the constraint violations.

Consider the equality-constrained problem

$$\min_x f(x) \text{ s.t. } c_i(x) = 0, \qquad i \in \mathcal{E}. \tag{1}$$

The quadratic penalty function $Q(x; \mu)$ for this formulation is

$$Q(x; \mu) \equiv f(x) + \frac{\mu}{2} c_i^2(x), \tag{2}$$

where $\mu > 0$ is the *penalty parameter*. By driving μ to ∞, we penalize the constraint violations with increasing severity.

Example

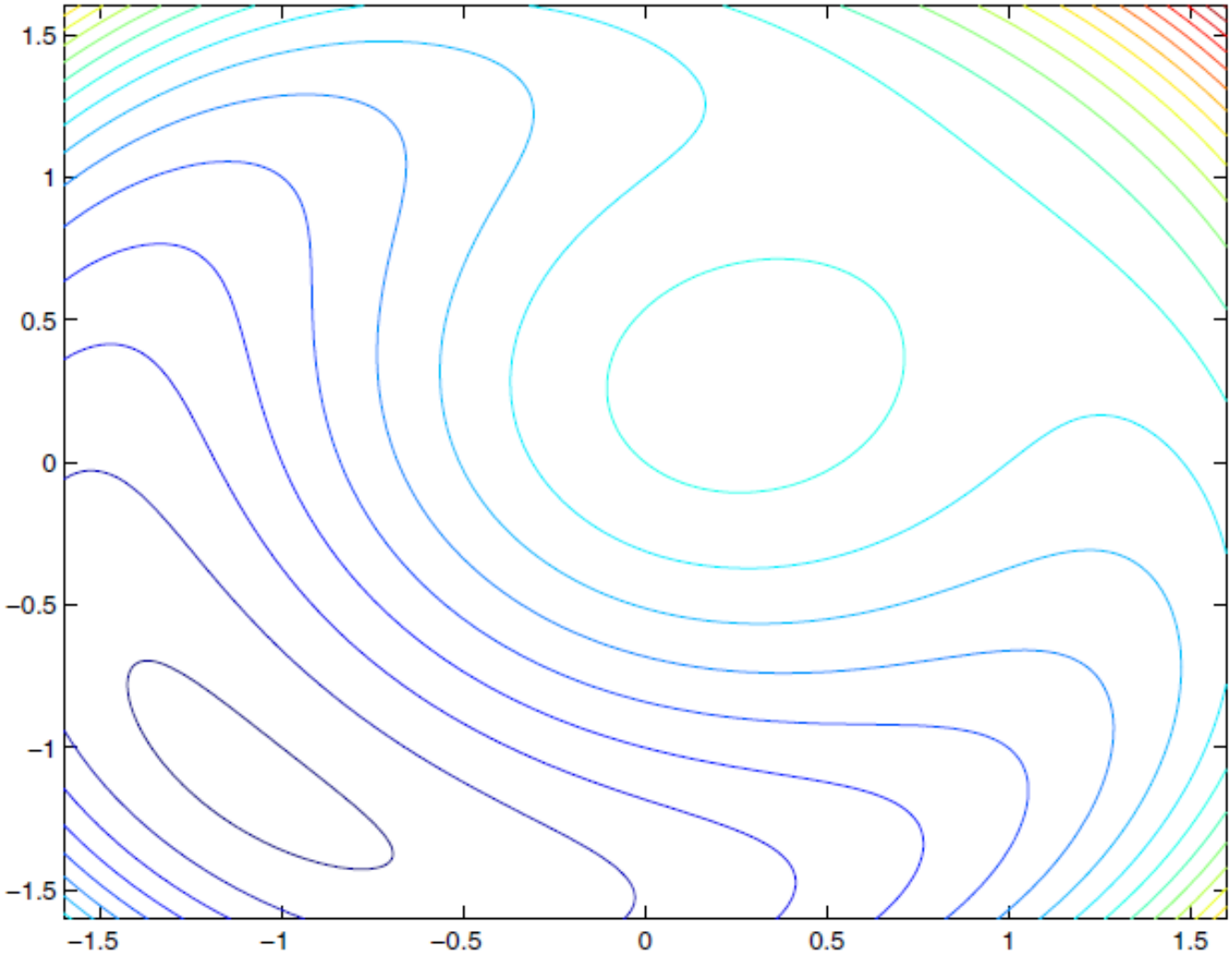
Consider the problem

$$\min_x x_1 + x_2 \text{ s.t. } x_1^2 + x_2^2 - 2 = 0,$$

for which the solution is $(-1, -1)^T$ and the quadratic function is

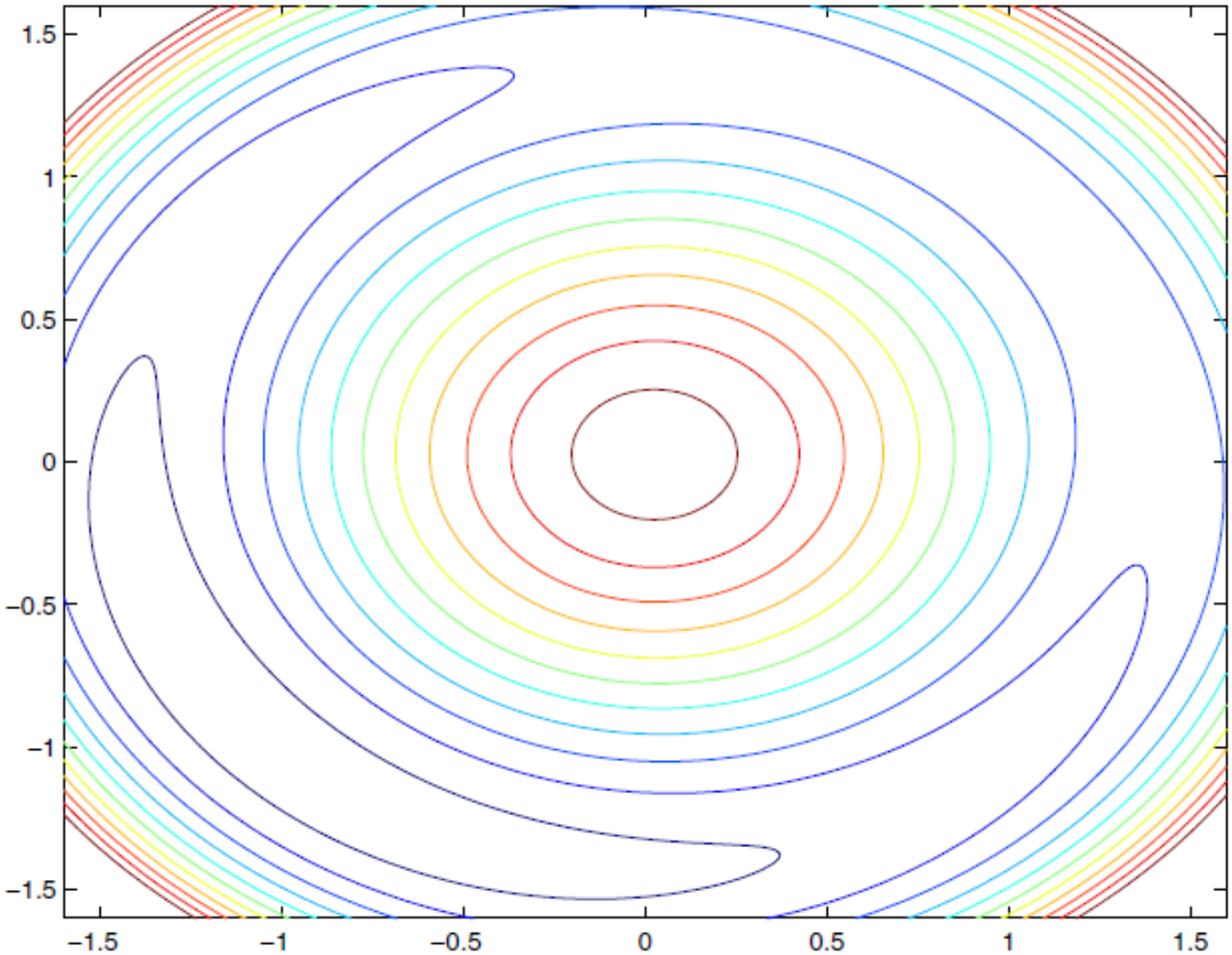
$$Q(x; \mu) = x_1 + x_2 + \frac{\mu}{2}(x_1^2 + x_2^2 - 2)^2.$$

Demo



Contours of $Q(x;\mu)$ for $\mu = 1$, contour spacing 0.5.

Demo



Contours of $Q(x;\mu)$ for $\mu = 10$, contour spacing 2.

Motivation

It makes good intuitive sense to consider a sequence of values $\{\mu_k\}$ with $\mu_k \uparrow \infty$ as $k \to \infty$, and to seek the approximate minimizer x_k of $Q(x; \mu_k)$ for each k.

Because the penalty terms in (2) are smooth, we can use techniques from unconstrained optimization to search for x_k. In search for x_k, we can use the minimizers x_{k-1}, x_{k-2}, etc., of $Q(\cdot, \mu)$ for smaller values of μ to construct an initial guess.

For suitable choices of the sequence $\{\mu_k\}$ and the initial guesses, just a few steps of unconstrained minimization may be needed for each μ_k.

The Quadratic Penalty Function

Consider the general constrained problem

$$\min_x f(x) \text{ s.t. } c_i(x) = 0,\ i \in \mathcal{E},\ c_i(x) \geq 0,\ i \in \mathcal{I}. \tag{3}$$

The quadratic penalty functions are defined as

$$Q(x;\mu) \equiv f(x) + \frac{\mu}{2} c_i^2(x) + \frac{\mu}{2} \sum_{i \in \mathcal{I}} ([c_i(x)]^-)^2, \tag{4}$$

where $[y]^-$ denotes $\max(-y, 0)$. In this case, Q may be less smooth than the objective and constraint functions. For instance, if one of the inequality constraints is $x_1 \geq 0$, then the function $\min(0, x_1)^2$ has a discontinuous second derivative, so that Q is no longer twice continuously differentiable.

Algorithmic Framework

Algorithm 1: Algorithmic Framework for Quadratic Penalty Method

Given $\mu_0 > 0$, a nonnegative sequence $\{\tau_k\}$ with $\tau_k \to 0$, and a starting point x_0^s;

for $k = 0, 1, 2, \cdots$

Find an approximate minimizer x_k of $Q(\cdot; \mu_k)$, starting at x_k^s, and terminating when $\|\nabla_x Q(x; \mu_k)\| \leq \tau_k$;

if final convergence test satisfied

stop with approximate solution x_k;

end(if)

Choose new penalty parameter $\mu_{k+1} > \mu_k$;

Choose new starting point x_{k+1}^s;

end(for)

Update strategy for parameter sequences $\{\mu_k\}$ and $\{\tau_k\}$

The parameter sequence $\{\mu_k\}$ can be chosen adaptively, based on the difficulty of minimizing the penalty function at each iteration.

- When minimization of $Q(x;\mu_k)$ proves to be expensive for some k, we choose μ_{k+1} to be only modestly larger than μ_k; for instance $\mu_{k+1} = 1.5\mu_k$.
- If we find the approximate minimizer of $Q(x;\mu_k)$ cheaply, we could try a more ambitious reduction, for instance $\mu_{k+1} = 10\mu_k$.

The convergence theory for Algorithm 1 allows wide latitude in the choice of tolerances τ_k; it requires only that $\tau_k \to 0$, to ensure that the minimization is carried out more and more accurately.

Convergence of the Quadratic Penalty Method

Theorem

Suppose that each x_k is the exact global minimizer of $Q(x;\mu_k)$ in above algorithm, and that $\mu_k \uparrow \infty$. Then every limit point x^* of the sequence $\{x_k\}$ is a solution of the problem (1).

Since this result requires us to find the *global* minimizer for each subproblem, its very desirable property of convergence to the global solution of (1) may be difficult to realize in practice.

The next result concerns convergence properties of the sequence $\{x_k\}$ when we allow inexact (but increasingly accurate) minimizations of $Q(\cdot;\mu_k)$.

Convergence of the Quadratic Penalty Method

In contrast to the previous theorem, it shows that the sequence is attracted to KKT points (that is, points satisfying first-order necessary conditions), rather than to a global minimizer. It also shows that the quantities $\mu_k c_i(x_k)$ may be used as estimates of the Lagrange multipliers λ_i in certain circumstances.

Theorem

Suppose that the tolerances and penalty parameters in above algorithm framework satisfy $\tau_k \to 0$ and $\mu_k \uparrow \infty$. Then if a limit point x^* of the sequence $\{x_k\}$ is infeasible, it is a stationary point of the function $\|c(x)\|^2$. On the other hand, if a limit point x^* is feasible and the constraint gradients $\nabla c_i(x^*)$ are linearly independent, then x^* is a KKT point for the porblem (1). For such points, we have for any infinite subsequence $\mathcal{K}$ such that $\lim_{k\in\mathcal{K}} x_k = x^*$ that

$$\min_{k\in\mathcal{K}} -\mu_k c_i(x_k) = \lambda_i^*, \qquad \forall i \in \mathcal{E},$$

where λ^* is the multiplier vector that satisfies the KKT conditions for the equality-constrained problem (1).

How to solve the unconstrained subproblem?

When only equality constraints are present, $Q(x;\mu_k)$ is smooth, so the algorithms for unconstrained minimization can be used to identify the approximate solution x_k. However, the minimization of $Q(x;\mu_k)$ becomes more and more difficult to perform when μ_k becomes large, unless we use special techniques to calculate the search directions.

For one thing, the Hessian $\nabla^2_{xx}Q(x;\mu_k)$ becomes quite ill conditioned near the minimizer. This property alone is enough to make unconstrained minimization algorithms such as quasi-Newton and conjugate gradient perform poorly.

How to solve the unconstrained subproblem?

Newton's method, on the other hand, is not sensitive to ill conditioning of the Hessian, but it, too, may encounter difficulties for large μ_k for two other reasons.

- First, ill conditioning of $\nabla^2_{xx} Q(x; \mu_k)$ might be expected to cause problems when we come to solve the linear equations to calculate the Newton step. We discuss this issue further at the end of the section, where we show that these effects are not so severe and that a reformulation of the Newton equations is possible.
- Second, even when x is close to the minimizer of $Q(x; \mu_k)$, the quadratic Taylor series approximation to $Q(x; \mu_k)$ about x is a reasonable approximation of the true function only in a small neighborhood of x. Since Newton's method is based on the quadratic model, the steps that it generates may not make rapid progress toward the minimizer of $Q(x; \mu_k)$, unless we are quite close to the minimizer. This difficulty can be overcome partly by judicious choice of the starting point x^s_{k+1}.

The Hessian is given by the formula

$$\nabla^2_{xx} Q(x;\mu_k) = \nabla^2 f(x) + \sum_{i\in\mathcal{E}} \mu_k c_i(x)\nabla^2 c_i(x) + \mu_k A(x)^T A(x), \tag{5}$$

where $A(x)^T = [\nabla c_i(x)]_{i\in\mathcal{E}}$. When x is close to the minimizer of $Q(\cdot;\mu_k)$ and the conditions of the above theorem is satisfied, we know that the sum of the first two terms is approximately equal to the Hessian of the Lagrangian function (2). To be specific, we have

$$\nabla^2_{xx} Q(x;\mu_k) \approx \nabla^2_{xx}\mathcal{L}(x,\lambda^*) + \mu_k A(x)^T A(x), \tag{6}$$

where x is chosen to the minimizer of $Q(\cdot,\mu_k)$. We see from this expression that $\nabla^2_{xx} Q(x;\mu_k)$ is approximately equal to the sum of

- a matrix whose elements are independent of μ_k (the Lagrangian term), and
- a matrix of rank $|\mathcal{E}|$ whose nonzero eigenvalues are of order μ_k (the summation term in (6)).

Ill Conditioning and Reformulations

The number of constraints $|\mathcal{E}|$ is usually fewer than n. In this case, the summation term is singular, and the overall matrix has some of its eigenvalues approaching a constant, while others are of order μ_k. Since μ_k is approaching infinity, the increasing ill conditioning of $Q(x;\mu_k)$ is apparent.
One consequence of the ill conditioning is possible inaccuracy in the calculation of the Newton step for $Q(x;\mu_k)$, which is obtained by solving the following system:

$$\nabla^2_{xx} Q(x;\mu_k)p = -\nabla_x Q(x;\mu_k). \tag{7}$$

Significant roundoff errors will appear in p regardless of the solution technique used, and algorithms will break down as the matrix becomes numerically singular. For the same reason, iterative methods can be expected to perform poorly unless accompanied by a preconditioning strategy that removes the systematic ill conditioning. The presence of roundoff error may not disqualify p from being a good direction of progress for Newton's method.

There is an alternative formulation of the equations (7) that avoids the ill conditioning due to the final term in (5). By introducing a new variables ζ defined by $\zeta = \mu A(x)p$, we see that the vector p that solves (7) also satisfies the following system:

$$\begin{bmatrix} \nabla^2 f(x) + \sum_{i\in\mathcal{E}} \mu_k c_i(x)\nabla^2 c_i(x) & A(x)^T \\ A(x) & -(1/\mu_k)I \end{bmatrix} \begin{bmatrix} p \\ \zeta \end{bmatrix} = \begin{bmatrix} -\nabla_x Q(x;\mu_k) \\ 0 \end{bmatrix}. \quad (8)$$

When x is not too far from the solution x^*, the coefficient matrix in this system does not have large values (of order μ_k), so the system (8) can be viewed as a well conditioned reformualtion of (7).

The formulation (8) allows us the view the quadratic penalty method either as the application of unconstrained minimization to the penalty method either as the application of unconstrained minimization to the penalty function $Q(\cdot, \mu_k)$ or as a variation on the SQP methods.

Outline

Exact Penalty Functions

Some penalty functions are *exact*, which means that, for certain choices of their penalty parameters, a single minimization with respect to x can yield the exact solution of the nonlinear programming problem. This property is desirable because it makes the performance of penalty methods less dependent on the strategy for updating the penalty parameter.

Exact Penalty Functions

Some penalty functions are *exact*, which means that, for certain choices of their penalty parameters, a single minimization with respect to x can yield the exact solution of the nonlinear programming problem. This property is desirable because it makes the performance of penalty methods less dependent on the strategy for updating the penalty parameter.

A popular nonsmooth penalty function for the general nonlinear programming problem (3) is the ℓ_1 *penalty function* defined by

$$\phi_1(x;\mu) = f(x) + \mu \sum_{i\in\mathcal{E}} |c_i(x)| + \mu \sum_{i\in\mathcal{I}} [c_i(x)]^-, \tag{9}$$

where $[y]^- = \max\{0, y\}$. Note that $\phi_1(x;\mu)$ is not differentiable at some x, because of the presence of the absolute value and $[\cdot]^-$ functions.

ℓ_1 Penalty Function

The following result establishes the *exactness* of the ℓ_1 penalty function.

Theorem

Suppose that x^* is a strict local solution of the nonlinear programming problem (3) at which the first-order necessary conditions are satisfied with Lagrangian multiplier λ_i^*, $i \in \mathcal{E} \cup \mathcal{I}$. Then x^* is a local minimizer of $\phi_1(x;\mu)$ for all $\mu > \mu^*$, where

$$\mu^* = \|\lambda^*\|_\infty = \max_{i \in \mathcal{E} \cup \mathcal{I}} |\lambda_i^*|. \tag{10}$$

If, in addition, the second-order sufficient conditions hold and $\mu > \mu^*$, then x^* is a strict local minimizer of $\phi_1(x;\mu)$.

Loosely speaking, at a solution of the nonlinear program x^*, any move into the infeasible region is penalized sharply enough that it produces an increase in the penalty function to a value greater than $\phi_1(x^*;\mu) = f(x^*)$, thereby forcing the minimizer of $\phi_1(\cdot,\mu)$ to lie at x^*.

ℓ_1 Penalty Function

Even though ϕ_1 is not differentiable, it has a directional derivative $D(\phi_1(x;\mu);p)$ along any direction.

Definition

A point $\hat{x} \in \Re^n$ is a stationary point for the penalty function $\phi_1(x;\mu)$ if

$$D(\phi_1(\hat{x};\mu)) \geq 0, \tag{11}$$

for all $p \in \Re^n$. Similarly, $\hat{x}$ is a stationary point of the measure of infeasibility

$$h(x) = \sum_{i\in\mathcal{E}} |c_i(x)| + \sum_{i\in\mathcal{I}} [c_i(x)]^- \tag{12}$$

if $D(h(\hat{x};p) \geq 0$ for all $p \in \Re^n$. If a point is infeasible for (3) but stationary with respect to the feasibility measure h, we say that it is an infeasible stationary point.

ℓ_1 Penalty Function

The following result complements the previous theory by showing that stationary point of $\phi_1(x;\mu)$ correspond to KKT points of the constrained optimization problem (3) under certain assumptions.

Theorem

Suppose that $\hat{x}$ is stationary point of the penalty function $\phi_1(x;\mu)$ for all μ greater than a certain threshold $\hat{\mu} > 0$. Then, if $\hat{x}$ is feasible for the nonlinear program (3), it satisfies the KKT conditions. If $\hat{x}$ is not feasible for (3), it is a stationary point of the measure of infeasibility.

Example

Consider the following problem in one variable:

$$\min x \text{ s.t. } x \geq 1.$$

whose solution is $x^* = 1$. We have that

$$\phi_1(x;\mu) = x + \mu[x-1]^- = \begin{cases} (1-\mu)x + \mu & \text{if } x \leq 1, \\ x & \text{if } x > 1. \end{cases}$$

The penalty function has a minimizer at x^* when $\mu > 1$, but is a monotone increasing function when $\mu < 1$.

Example

Consider again the problem

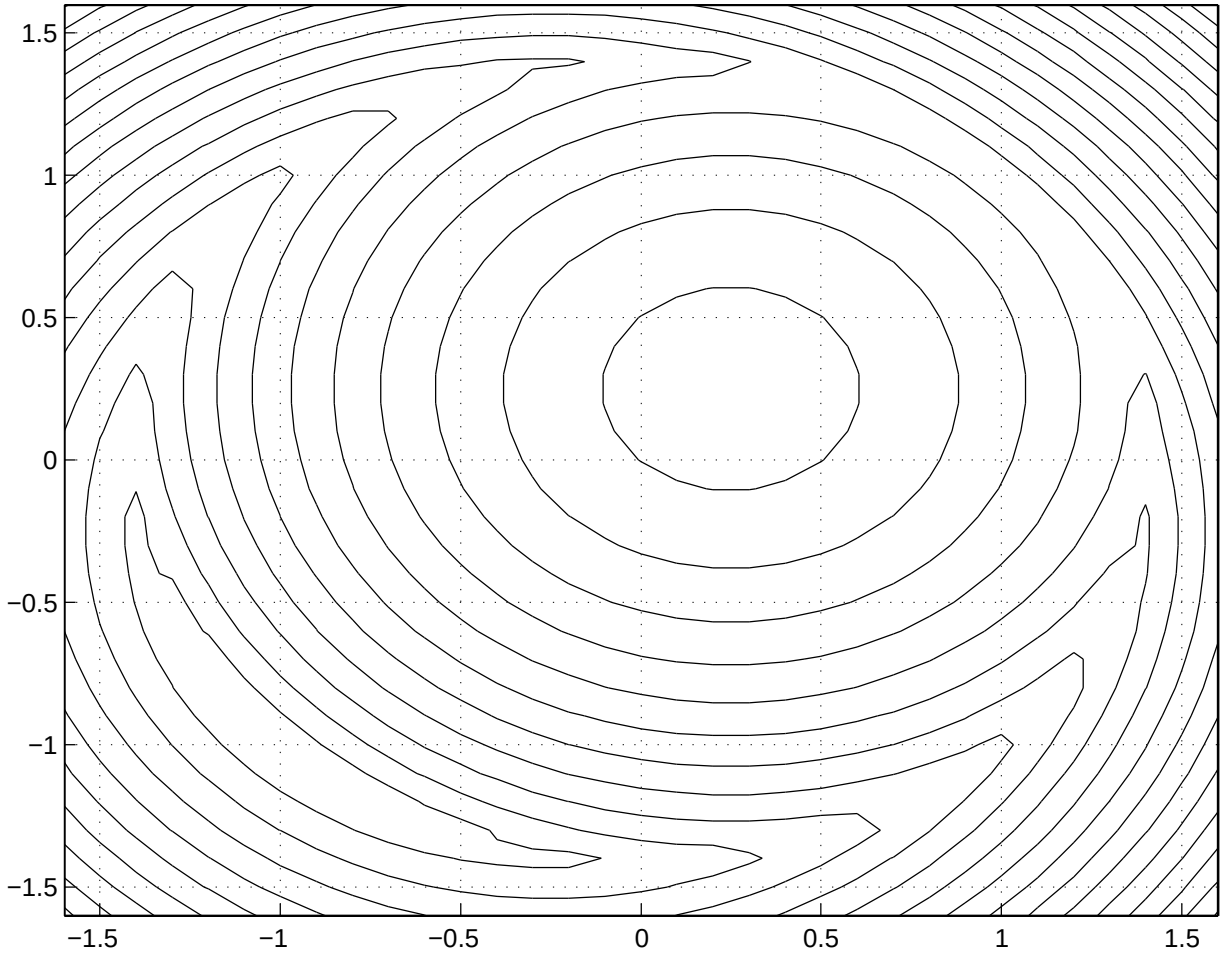
$$\min_x x_1 + x_2 \text{ s.t. } x_1^2 + x_2^2 - 2 = 0,$$

for which ℓ penalty function is

$$\phi_1(x;\mu) = x_1 + x_2 + \mu|x_1^2 + x_2^2 - 2|.$$

We find that for all $\mu > |\lambda^*| = 0.5$, the minimizer of $\phi_1(x;\mu)$ coincides with x^*.

Demo



Contours of $\phi_1(x;\mu)$ for $\mu = 2$, contour spacing 0.5. The sharp corners on the contours indicate nonsmoothness along the boundary of the circle defined by $x_1^2 + x_2^2 = 2$.

Algorithm 2: Classical ℓ_1 Penalty Method

Given $\mu_0 > 0$, tolerance $\tau > 0$, starting point x_0^s;
for $k = 0, 1, 2, \cdots$
    Find an approximate minimizer x_k of $\phi_1(x; \mu_k)$, starting at x_k^s;
    if $h(x_k) \leq \tau$
        stop with a approximate solution x_k;
    end(if)
    Choose new penalty parameter $\mu_{k+1} > \mu_k$;
    Choose new starting point x_{k+1}^s
end(for)

ℓ_1 Penalty Method

The simplest scheme for updating the penalty parameter μ_k is to increase it by a constant multiple (say 5 or 10), if the current value produces a minimizer that is not feasible to within the tolerance τ. This scheme sometimes works well in practice, but can also be inefficient.

- If the initial penalty parameter μ_k is too small, many cycles of the above algorithm may be needed to determine an appropriate value. In addition, the iterates may move away from the solution x^* in these initial cycles, in which case the minimization of $\phi_1(x, \mu_k)$ should be terminated early and x_k^s should possible be reset to a previous iterate.
- If, on the other hand, μ_k is excessively large, the penalty function will be difficult to minimize, possibly requiring a large number of iterations.

A General Class of Nonsmooth Penalty Methods

Exact nonsmooth penalty functions can be defined in terms of norms other than the ℓ_1 norm. We can write

$$\phi(x;\mu) = f(x) + \mu\|c_{\mathcal{E}}(x)\| + \mu\|[c_{\mathcal{I}}(x)]^-\|. \tag{13}$$

where $\|\cdot\|$ is any vector norm, and all the equality and inequality constraints have been grouped in the vector functions $c_{\mathcal{E}}$ and $c_{\mathcal{I}}$, respectively. The most common norms used in practice are the ℓ_1, ℓ_∞ and ℓ_2(not squared).

- The algorithm framework for ℓ_1 penalty function applies to any of these penalty functions; we simply redefine the measure of infeasibility as $h(x) = \|c_{\mathcal{E}}(x)\| + \|[c_{\mathcal{I}}(x)]^-\|$.
- The theoretical properties described for the ℓ_1 function extend to the general class (13) without modification, except that (10) is replaced by $\mu^* = \|\lambda^*\|_D$, where $\|\cdot\|_D$ is the dual norm of $\|\cdot\|$.

Outline

Motivation

We first consider the equality-constrained problem (1). The quadratic penalty function $Q(x;\mu)$ defined by (2) penalizes constraint violations by squaring the infeasibilities and scaling them by $\mu/2$. However, the approximate minimizers x_k of $Q(x;\mu_k)$ do not quite satisfy the feasibility conditions $c_i(x) = 0$, $i \in \mathcal{E}$. Instead, they are perturbed slightly to approximately satisfy

$$c_i(x_k) \approx -\lambda_i^*/\mu_k, \forall i \in \mathcal{E}. \tag{14}$$

To be sure, this perturbation vanishes as $c_i(x_k) \to 0$ as $\mu_k \uparrow \infty$, but one may ask whether we can alter the function $Q(x;\mu_k)$ to avoid this systematic perturbation - that is, to make the approximate minimizers more nearly satisfy the equality constraints $c_i(x) = 0$, even for moderate values of μ_k. By doing so, we may avoid the need to decrease μ to infinity, and thereby avoid the ill conditioning and numerical problems associated with $Q(x;\mu)$ for small values of this penalty parameter.

Motivation

The augmented Lagrangian function $\mathcal{L}_A(x, \lambda; \mu)$ achieves these goals by including an explicit estimate of the Lagrange multipliers λ, based on the formula (14), in the objective. From the definition

$$\mathcal{L}_A(x, \lambda; \mu) \equiv f(x) - \sum_{i \in \mathcal{E}} \lambda_i c_i(x) + \frac{\mu}{2} \sum_{i \in \mathcal{E}} c_i^2(x), \tag{15}$$

we see that the augmented Lagrangian differs from the (standard) Lagrangian for (1) by the presence of the squared terms, while it differs from the quadratic penalty function (2) in the presence of the summation term involving the λ. In this sense, it is a combination of the Lagrangian and quadratic penalty functions.

Motivation

We now design an algorithm that fixes the barrier parameter μ to some value $\mu_k > 0$ at its kth iteration, fixes λ at the current estimate λ_k, and performs minimization with respect to x. Using x_k to denote the approximate minimizer of $\mathcal{L}_A(x, \lambda^k; \mu_k)$, we have by the optimality conditions for unconstrained minimization that

$$0 \approx \nabla_x \mathcal{L}_A(x, \lambda^k; \mu_k) = \nabla f(x_k) - \sum_{i \in \mathcal{E}} [\lambda_i^k - \mu_k c_i(x_k)] \nabla c_i(x_k). \tag{16}$$

By comparing with the optimality condition for (1), we can deduce that

$$\lambda_i^* \approx \lambda_i^k - \mu_k c_i(x_k), \qquad \forall i \in \mathcal{E}. \tag{17}$$

By rearranging this expression, we have that

$$c_i(x_k) \approx -\frac{1}{\mu_k}(\lambda_i^* - \lambda_i^k), \qquad \forall i \in \mathcal{E},$$

so we conclude that if λ_k is close to the optimal multiplier vector λ^*, the infeasibility in x_k will be much smaller than $(1/\mu_k)$, rather than being proportional to $(1/\mu_k)$ as in (14).

Motivation

How can we update the multiplier estimates λ_k from iteration to iteration, so that they approximate λ^* more and more accurately, based on current information? Equation (17) immediately suggests the formula

$$\lambda_i^{k+1} = \lambda_i^k - \mu_k c_i(x_k), \qquad \forall i \in \mathcal{E} \tag{18}$$

Algorithm 3: Augmented Lagrangian Method

Given $\mu_0 > 0$, tolerance $\tau_0 > 0$, starting point x_0^s and λ^0;
for $k = 0, 1, 2, \cdots$
    Find an approximate minimizer x_k of $\mathcal{L}_A(\cdot, \lambda^k; \mu_k)$, starting at x_k^s;
        and terminating when $\|\nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k)\| \leq \tau_k$;
    if a convergence test for (1) is satisfied
        stop with approximate solution x_k;
    end(if)
    Update Lagrange multipliers using (18) to obtain λ^{k+1};
    Choose new penalty parameter $\mu_{k+1} > \mu_k$;
    Choose new starting point x_{k+1}^s
    Select tolerance τ_{k+1};
end(for)

Comments on the Algorithm

We show below that convergence of this method can be assured without increasing μ indefinitely. Ill conditioning is therefore less of a problem than in the framework of quadratic function, so the choice of starting point x_{k+1}^s in the above framework is less critical. (In fact, we simply start the search at iteration $k+1$ from the previous approximate minimizer x_k.)

The tolerance τ_k could be chosen to depend on the infeasibility $\sum_{i\in\mathcal{E}} |c(x_k)|$, and the penalty parameter μ may be increased if the reduction in this infeasibility measure is insufficient at the present iteration.

Example

Consider again the problem

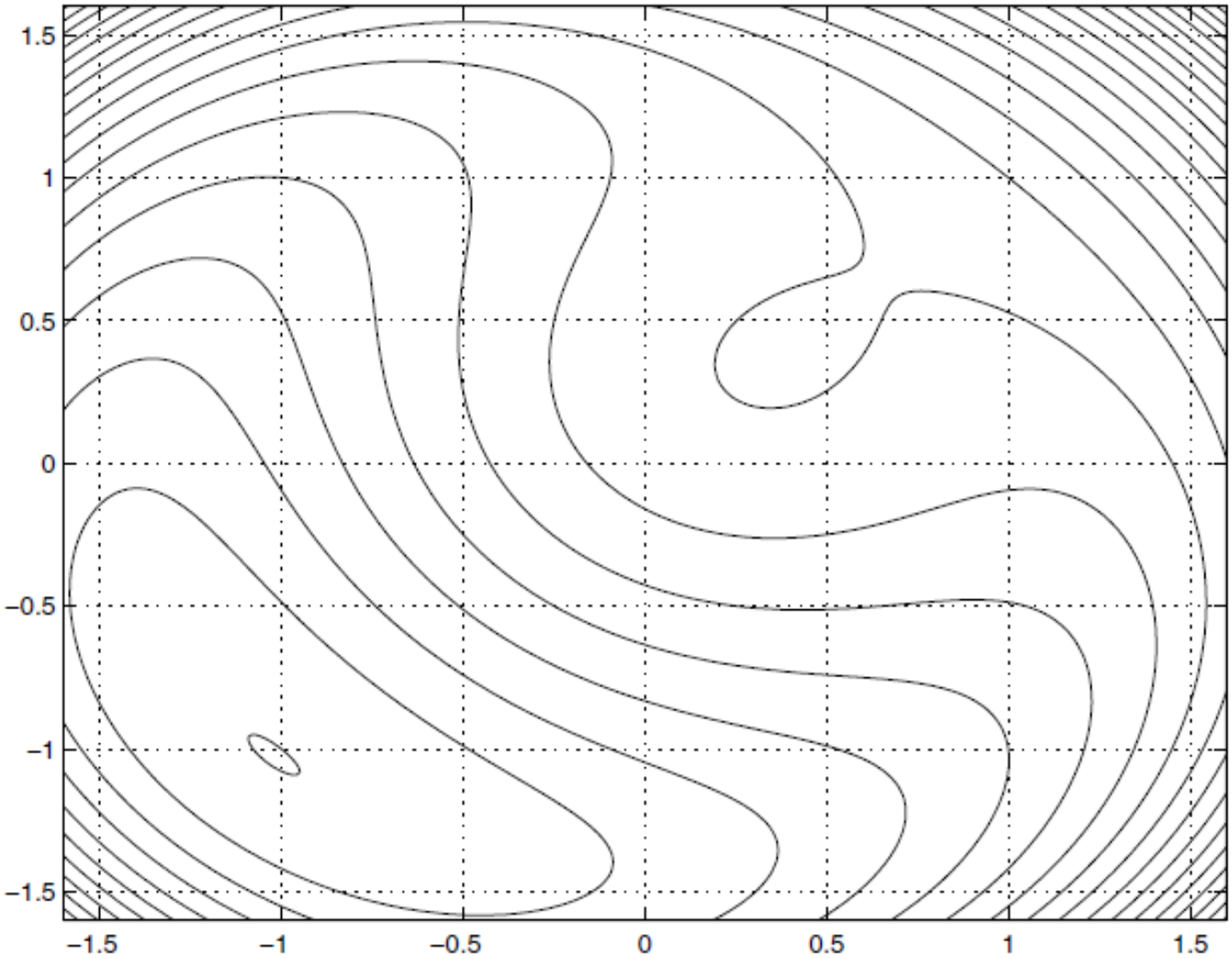
$$\min_x x_1 + x_2 \text{ s.t. } x_1^2 + x_2^2 - 2 = 0,$$

for which the solution is $x^* = (-1, -1)^T$ and the optimal Lagrangian multiplier is $\lambda^* = -0.5$. Consider the augmented Lagrangian

$$\mathcal{L}_A(x, \lambda; \mu) = x_1 + x_2 - \lambda(x_1^2 + x_2^2 - 2) + \frac{\mu}{2}(x_1^2 + x_2^2 - 2)^2.$$

Suppose that at iterate k we have $\mu_k = 1$, while the current multiplier estimate is $\lambda_k = -0.4$.

Demo



Contours of $\mathcal{L}_A(x, -0.4; 1)$, contour spacing 0.5.

Example

Note that the spacing of the contours indicates that the conditioning of this problem is similar to that of the quadratic penalty function $Q(x; 1)$. However, the minimizing value of $x_k \approx (-1.02, -1.02)$ is much closer to the solution $x^* = (-1, -1)^T$ than is the minimizing value of $Q(x; 1)$, which is approximately $(-1.1, -1.1)$. This example shows that the inclusion of the Lagrange multiplier term in the function $\mathcal{L}_A(x, \lambda; \mu)$ can result in a substantial improvement over the quadratic penalty method, as a way to reformulate the constrained optimization problem (3).

Properties of the Augmented Lagrangian

Theorem

Let x^* be a local solution of (1) at which the LICQ is satisfied (that is, the gradients $\nabla c_i(x^*)$, $i \in \mathcal{E}$, are linearly independent vectors), and the second-order sufficient conditions are satisfied for $\lambda = \lambda^*$. Then there is a threshold value $\bar{\mu}$ such that for all $\mu \geq \bar{\mu}$, x^* is a strict local minimizer of $\mathcal{L}_A(x, \lambda^*; \mu)$.

This result validates the approach of algorithm 3 by showing that when we have knowledge of the exact Lagrange multiplier vector λ^*, the solution x^* of (1) is a strict minimizer of $\mathcal{L}_A(x, \lambda^*; \mu)$ for all μ sufficiently small. Although we do not know λ^* exactly in practice, the result and its proof strongly suggest that we can obtain a good estimate of x^* by minimizing $\mathcal{L}_A(x, \lambda; \mu)$ even when μ is not particularly large, provided that λ is a reasonable estimate of λ^*.

Properties of the Augmented Lagrangian

The second result, given by Bertsekas, describes the more realistic situation of $\lambda \neq \lambda^*$. It gives conditions under which there is a minimizer of $\mathcal{L}_A(x, \lambda; \mu)$ that lies close to x^* and gives error bounds on both x_k and the updated multiplier estimate λ_{k+1} obtained from solving the subproblem at iteration k.

Theorem

Suppose that the assumptions of above theorem are satisfied at x^* and λ^*, and let $\bar{\mu}$ be chosen as in that theorem. Then there exist positive scalars δ, ϵ, and M such that the following claims hold:
(a) For all λ_k and μ_k satisfying

$$\|\lambda^k - \lambda^*\| \leq \mu_k \delta, \qquad \mu_k \geq \bar{\mu}, \tag{19}$$

the problem

$$\min_x \mathcal{L}_A(x, \lambda_k; \mu_k) \ s.t. \|x - x^*\| \leq \epsilon$$

has a unique solution x_k. Moreover, we have

$$\|x_k - x^*\| \leq M\|\lambda^k - \lambda^*\|/\mu_k. \tag{20}$$

Properties of the Augmented Lagrangian

Theorem

(b) For all λ_k and μ_k that satisfy (19), we have

$$\|\lambda^{k+1} - \lambda^*\| \leq M\|\lambda^k - \lambda^*\|/\mu_k, \tag{21}$$

where λ^{k+1} is given by the formula (18).
(c) For all λ^k and μ_k that satisfy (19), the matrix $\nabla^2_{xx}\mathcal{L}_A(x_k, \lambda^k; \mu_k)$ is positive definite and the constraint gradients $\nabla c_i(x_k)$, $i \in \mathcal{E}$, are linearly independent.

Properties of the Augmented Lagrangian

This theorem illustrates some salient properties of the augmented Lagrangian approach.

- The bound (20) shows that x_k will be close to x^* if λ_k is accurate *or* if the penalty parameter μ_k is large. Hence, this approach gives us two ways of improving the accuracy of x_k, whereas the quadratic penalty approach gives us only one option: increasing μ_k.
- The bound (21) states that, locally, we can ensure an improvement in the accuracy of the multiplier by choosing a sufficient large value of μ_k.
- The final observation of the theorem shows that second-order sufficient conditions for unconstrained minimization are also satisfied for the kth subproblem under the given conditions, so one can expect good performance by applying standard unconstrained minimization techniques.

Outline

Bound-Constrained Formulation

Given the general nonlinear program (3), we can convert it to a problem with equality constraints and bound constraints by introducing slack variables s_i and replacing the general inequalities $c_i(x) \geq 0$, $i \in \mathcal{I}$, by

$$c_i(x) - s_i = 0, \qquad s_i \geq 0, \qquad \forall i \in \mathcal{I}. \tag{22}$$

Bound constraints, $l \leq x \leq u$, need not to be transformed. By reformulating in this way, we can write the nonlinear program as follows:

$$\min_{x \in \Re^n} f(x) \text{ s.t. } c_i(x) = 0, i = 1, 2, \cdots, m, l \leq x \leq u. \tag{23}$$

Some of the components of the lower bound vector l may be set to $-\infty$, signifying that there is no lower bound on the components of x in question; similarly for μ.

Bound-Constrained Lagrangian

The bound-constrained Lagrangian (BLC) approach incorporates only the equality constraints from (23) into the augmented Lagrangian, that is,

$$\mathcal{L}_A(x, \lambda; \mu) = f(x) - \sum_{i=1}^{m} \lambda_i c_i(x) + \frac{\mu}{2} \sum_{i=1}^{m} c_i^2(x). \tag{24}$$

The bound constraints are enforced explicitly in the subproblem, which has the form

$$\min_x \mathcal{L}_A(x, \lambda; \mu) \text{ s.t } l \leq x \leq u. \tag{25}$$

After this problem has been solved approximately, the multipliers λ and the penalty parameter μ are updated and the process is repeated.

Subproblem Solution

An efficient techniques for solving the nonlinear program with bound constraints (25) (for fixed μ and λ) is the (nonlinear) gradient projection method. By specializing the KKT conditions to the problem (25), we find that the first-order necessary condition for x to be a solution of (25) is that

$$x - P(x - \nabla_x \mathcal{L}_A(x, \lambda, \mu), l; u) = 0, \tag{26}$$

where $P(g, l, u)$ is the projection of the vector $g \in \Re^n$ onto the rectangular box $[l, u]$ defined as follows

$$P(g, l, u) = \begin{cases} l_i & \text{if } g_i \leq l_i, \\ g_i & \text{if } g_i \in (l_i, u_i), \\ u_i & \text{if } g_i \geq u_i. \end{cases} \qquad \forall i = 1, 2, \cdots, n. \tag{27}$$

Algorithm 4: Bound-Constrained Lagrangian Method

Given an initial point x_0 and initial multiplier λ^0;
Choose convergence tolerances η_* and ω_*;
Set $\mu_0 = 10$, $\omega_0 = 1/\mu_0$, and $\eta_0 = 1/\mu_0^{0.1}$;
for $k = 0, 1, 2, \cdots$

Find an approximate minimizer x_k of the subproblem (25) such that

$$\|x_k - P(x_k - \nabla_x \mathcal{L}_A(x_k, \lambda^k, \mu_k), l, u)\| \leq \omega$$

if $\|c(x_k)\| \leq \eta_k$

(* test for convergence *)

if $\|c(x_k)\| \leq \eta_*$ and $\|x_k - P(x_k - \nabla_x \mathcal{L}_A(x_k, \lambda^k, \mu_k), l, u)\| \leq \omega_k$

stop with approximate solution x_k;

end(if)

(* Update Lagrange multipliers, tighten tolerances *)

$\lambda^{k+1} = \lambda^k - \mu_k c(x_k)$; $\mu_{k+1} = \mu_k$; $\eta_{k+1} = \eta_k/\mu_{k+1}^{0.9}$; $\omega_{k+1} = \omega_k/\mu_{k+1}$;

else

(* increase penalty parameter, tighten tolerances *)

$\lambda^{k+1} = \lambda^k$; $\mu_{k+1} = 100\mu_k$; $\eta_{k+1} = 1/\mu_{k+1}^{0.1}$; $\omega_{k+1} = 1/\mu_{k+1}$;

end(if)

end(for)

Linearly Constrained Formulation

The principal idea behind *linearly constrained Lagrangian* (LCL) methods is to generate a step by minimizing the Lagrangian (or augmented Lagrangian) subject to linearizations of the constraints. If we use the formulation (23) of the nonlinear programming problem, the subproblem used in the LCL approach takes the form

$$\min_x \quad F_k(x) \tag{28a}$$

$$s.t. \quad c(x_k) + A_k(x - x_k) = 0, \quad l \leq x \leq u. \tag{28b}$$

There are several possible choices for $F_k(x)$. Current LCL methods define F_k to be the augmented Lagrangian function

$$F_k(x) = f(x) - \sum_{i=1}^{m} \lambda_i^k \bar{c}_i^k(x) + \frac{\mu}{2} \sum_{i=1}^{m} [\bar{c}_i^k(x)]^2. \tag{29}$$

where $\bar{c}_i^k(x)$ is the difference between $c_i(x)$ and its linearization at x_k, that is,

$$\bar{c}_i^k(x) = c_i(x) - c_i(x_k) - \nabla c_i(x_k)^T (x - x_k). \tag{30}$$

Unconstrained Formulation

We can obtain an unconstrained form of the augmented Lagrangian subproblem for inequality-constrained problems by using a derivation based on the proximal point approach. Suppose for simplicity that the problem has non equality constraints ($\mathcal{E} = \emptyset$), we can write the problem (3) equivalently as an unconstrained optimization problem:

$$\min_{x \in \Re^n} F(x), \tag{31}$$

where

$$F(x) = \max_{\lambda \geq 0} \left\{ f(x) - \sum_{i \in \mathcal{I}} \lambda_i c_i(x) \right\} = \begin{cases} f(x) & \text{if x is feasible} \\ \infty & \text{otherwise.} \end{cases} \tag{32}$$

By combining (31) and (32), we have

$$\min_{x \in \Re^n} F(x) = \min_{x \text{ feasible}} f(x), \tag{33}$$

which is simply the original inequality-constrained problem.

Unconstrained Formulation

We can make this approach more practical by replacing F by a smooth approximation $\hat{F}(x;\lambda^k,\mu_k)$ which depends on the penalty parameter μ_k and Lagrange multiplier estimate λ^k. This approximation is defined as follows:

$$\hat{F}(x;\lambda^k,\mu_k) = \max_{\lambda\geq 0}\{f(x) - \sum_{i\in\mathcal{I}}\lambda_i c_i(x) - \frac{1}{2\mu_k}\sum_{i\in\mathcal{I}}(\lambda_i-\lambda_i^k)^2\}. \tag{34}$$

The final term in this expression applies a penalty for any move of λ away from the previous estimate λ^k; it encourages the new maximizer λ to stay *proximal* to the previous estimate λ^k. Since (34) represents a bound-constrained quadratic problem in λ, separable in the individual components λ_i, we can perform the maximization explicitly, to obtain

$$\lambda_i = \begin{cases} 0 & \text{if } -c_i(x)+\lambda_i^k/\mu_k \leq 0; \\ \lambda_i^k - \mu_k c_i(x) & \text{otherwise.} \end{cases} \tag{35}$$

Unconstrained Formulation

By substituting these values in (34), we find that

$$\hat{F}(x;\lambda^k,\mu_k) = f(x) + \sum_{i\in\mathcal{I}} \psi(c_i(x),\lambda_i^k;\mu_k), \tag{36}$$

where the function ψ of three scalar arguments is defined as follows:

$$\psi(t,\sigma;\mu) \equiv \begin{cases} -\sigma t + \frac{\mu}{2}t^2 & \text{if } t - \sigma/\mu \leq 0, \\ -\frac{1}{2\mu}\sigma^2 & \text{otherwise,} \end{cases} \tag{37}$$

Hence, we can obtain the new iterate x_k by minimizing $\hat{F}(x;\lambda^k,\mu_k)$ with respect to x, and use the formula (35) to obtain the updated Lagrange multiplier estimate λ^{k+1}.

Outline

Perspective and Software

Augmented Lagrangian methods have been popular for many years because, in part, of their simplicity. The MINOS and LANCELOT packages rank among the best implementation of augmented Lagrangian methods. Both are suitable for large-scale nonlinear programming problems. At a general level, the linearly constrained Lagrangian of MINOS and the bound-constrained Lagrangian method of LANCELOT have important features in common. They differ significantly, however, in the formulation of the step computation subproblems and in the techniques used to solve these subproblems.

- MINOS follows a reduced-space approach to handle linearized constraints and employs a (dense) quasi-Newton approximation to the Hessian of the Lagrangian. As a result, Minos is most successful for problems with relatively few degrees of freedom.
- LANCELOT, on the other hand, is more effective when there are relatively few constraints. LANCELOT does not require a factorization of the constraint Jacobian matrix A, again enhancing its suitability for very large problems, and provides a variety of Hessian approximation options and preconditioners.

Conclusion

An important class of methods for constrained optimization seeks the solution by replacing the original constrained problem by a sequence of unconstrained subproblems.

- Inexact Penalty Function Approaches:
  - The *quadratic penalty* method replaces the constraints by penalty terms in the objective function, where each penalty term is a multiple of the square of the constraint violation.
  - The *log-barrier method*, in which logarithmic terms prevent feasible iterates from moving too close to the boundary of the feasible region, is interesting in its own right, and it also provides the foundation of primal and primal-dual interior-point methods, which have proved to be important in LP, convex QP, and SQP, and may yet make their mark in general constrained optimization as well.

Conclusion

- Exact Penalty Function Approaches:
  - In *nonsmooth exact penalty* methods, a single unconstrained problem (rather than a sequence) takes the place of the original constrained problem. Using these penalty functions, we can often find a solution by performing a single unconstrained minimization, but the nonsmoothness may create complications. A popular function of this type is the ℓ_1 penalty function.
  - A different kind of exact penalty approach is the *method of multipliers* or *augmented Lagrangian method*, in which explicit Lagrange multiplier estimates are used to avoid the ill-conditioning that is inherent in the quadratic penalty function.

Thanks for your attention!