Introduction to inequality-constrained optimization: The logarithmic barrier method

Mark S. Gockenbach

1 Introduction

I now turn to the inequality-constrained nonlinear program

$$\begin{aligned} \min \quad & f(x) && (1)\\ s.t. \quad & h(x) \geq 0, && (2) \end{aligned}$$

where $f : \mathrm{R}^n \to \mathrm{R}$ and $h : \mathrm{R}^n \to \mathrm{R}^p$. The reader should notice that $h(x) \geq 0$ is to be interpreted component-wise; that is, $h(x) \geq 0$ if and only if $h_i(x) \geq 0$ for each $i = 1, 2, \ldots, p$. Inequality constraints are common in optimization problems; indeed, almost every optimization problems does or could have inequality constraints, although it is sometimes safe to ignore them.

The most common type of inequality constraints are *simple bounds* that the variables must satisfy. For example, if every variable must satisfy both lower and upper bound, then the constraints could be written as

$$a \leq x \leq b,$$

where $a, b \in \mathrm{R}^n$ with $a \leq b$. To incorporate these constraints into the standard form (2), one would write them as

$$\begin{aligned} x - a &\geq 0,\\ b - x &\geq 0. \end{aligned}$$

Therefore, if a problem contained only these simple bounds, the constraint function $h : \mathrm{R}^n \to \mathrm{R}^p$ ($p = 2n$) would be defined as

$$h(x) = \begin{bmatrix} x - a \\ b - x \end{bmatrix} = \begin{bmatrix} x_1 - a_1 \\ x_2 - a_2 \\ \vdots \\ x_n - a_n \\ b_1 - x_1 \\ b_2 - x_2 \\ \vdots \\ b_n - x_n \end{bmatrix}.$$

Simple bounds are common because variables typically have a physical interpretation and some real numbers do not make physical sense for a given variable. For example, physical parameters (density, elasticity, thermal conductivity, etc.) typically cannot take negative values, so *nonnegativity constraints* of the form

$$x \geq 0$$

are common. The same constraint appears when the variables represents quantities (for example, the number of barrels of petroleum) that cannot be negative. Upper bounds often represent limited resources.

In some cases, it may seem that the appropriate constraint is a strict inequality, as when the variables represent physical parameters that must be strictly positive. However, a strict inequality constraint may lead to an optimization problem that is ill-posed in that a minimizer is infeasible but on the boundary of the feasible set. In such as case, there may not be a solution to the optimization problem. A simple example of this is

$$\begin{array}{ll} \min & x^2 \\ s.t. & x > 0. \end{array}$$

For this reason, strict inequality constraints are not used in nonlinear programming.

When the appropriate constraint seems to be a strict inequality, one of the following is usually true:

1. The problem is expected to have a solution that easily satisfies the strict inequality. In this case, as I will show below, the constraint plays little part in the theory or algorithm other than as a "sanity check" on the variable. Therefore, the nonstrict inequality constraint is just as useful.

2. Due to noise in the data or other complications, the solution to the optimization problem may lie on the boundary of the feasible set, even though such a solution is not physically meaningful. In this case, the inequality must be perturbed slightly and written as a nonstrict inequality. For example, a constraint of the form $x_i > 0$ should be replaced with $x_i \geq a_i$, where $a_i > 0$ is the smallest value of x_i that is physically meaningful.

It may not be clear how to distinguish the first case from the second; if it is not, the second approach is always valid. The first case can usually be identified by the fact that the solution is not expected to be close to satisfying the inequality as an equation.

Inequality constraints that are not simple bounds are usually bounds on derived quantities. For example, if x represents design parameters for a certain object, the mass m of the object may be represented as a function of x: $m = q(x)$. In this case, constraints of the form $q(x) \geq M_1$ or $q(x) \leq M_2$ (or both) may be appropriate.

I now present a simple example of an inequality-constrained nonlinear program.

Example 1.1 *I define* $f : \mathrm{R}^2 \to \mathrm{R}$ *and* $h : \mathrm{R}^2 \to \mathrm{R}^2$ *by*

$$f(x) = (x_1 - 1)^2 + 2(x_2 - 2)^2, \;\; h(x) = \begin{bmatrix} 1 - x_1^2 - x_2^2 \\ x_1 + x_2 \end{bmatrix}$$

The feasible set for

$$\begin{array}{ll} \min & f(x) \\ s.t. & h(x) \geq 0 \end{array}$$

is shown in Figure 1, together with the contours of the objective function f.

An important aspect of this example is that the second constraint, $x_1 + x_2 \geq 0$, *does not affect the solution. If the second constraint were changed to* $x_1 + x_2 \geq u$ *for some value of* u *that is not too large, or if the second constraint were simply omitted, the optimization problem would have the same solution. Both the theory of and algorithms for inequality-constrained problems must address the issue of "inactive" constraints.*

The first step in analyzing NLP (1–2) should be to derive the optimality conditions. However, since the optimality conditions are somewhat more complicated than in the case of equality constraints, I will begin by presenting an algorithm for solving (1–2), namely, the logarithmic barrier method. From this algorithm, I will deduce the optimality conditions for the inequality-constrained NLP. A rigorous derivation of these optimality conditions will be given later.

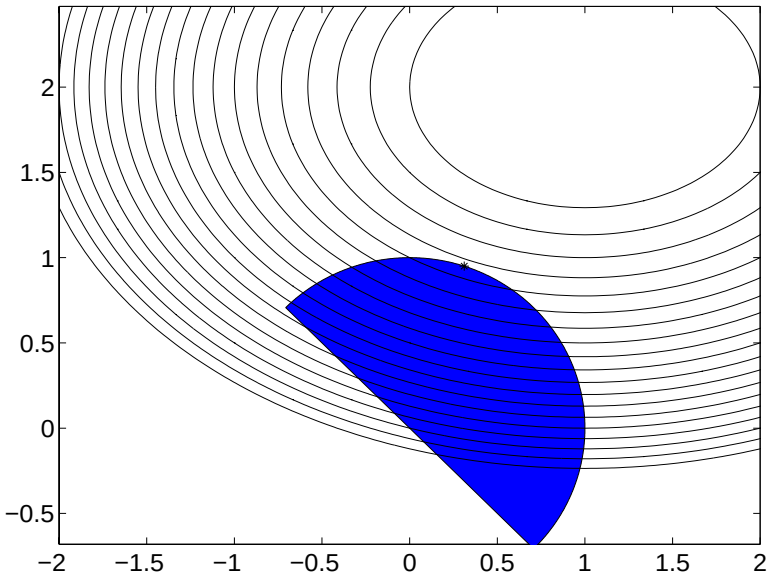

Figure 1: The contours of f and the feasible set determined by $h(x) \geq 0$ (see Example 1.1). The feasible set is half of the unit disk (the shaded region). The minimizer $x^* \doteq (0.3116, 0.9502)$ is indicated by an asterisk.

2 The logarithmic barrier method

Assuming it is possible to find a *strictly feasible* point $x^{(0)}$, that is, a point satisfying $h(x^{(0)}) > 0$, a natural strategy for solving (1–2) is to decrease f as much as possible while ensuring that the boundary of the feasible set is never crossed. One way to prevent an optimization algorithm from crossing the boundary is to assign a penalty to approaching it. The most popular way of doing this is to augment the objective function by a *logarithmic barrier* term:

$$B(x; \mu) = f(x) - \mu \sum_{i=1}^{p} \log\left(h_i(x)\right). \tag{3}$$

Here log denotes the natural logarithm. Since

$$-\log(t) \to \infty \text{ as } t \to 0,$$

$B(\cdot; \mu)$ "blows up" at the boundary and therefore presents an optimization algorithm with a "barrier" to crossing the boundary. Of course, the solution to an inequality-constrained is likely to lie on the boundary of the feasible set, so the barrier must be gradually removed by reducing μ toward zero. This suggests the following strategy:

> Choose $\mu_0 > 0$ and a strictly feasible point $x^{(0)}$.
>
> For $k = 1, 2, 3, \ldots$
>
> > Choose $\mu_k \in (0, \mu_{k-1})$ (perhaps $\mu_k = \beta\mu_{k-1}$ for some constant $\beta \in (0, 1)$).
> > Using $x^{(k-1)}$ as the starting point, solve
> >
> > $$\min B(x; \mu_k)$$
> >
> > to get $x^{(k)}$.

μ	$\|x^* - x^*_\mu\|$
1	$1.5912 \cdot 10^{-1}$
2^{-1}	$9.4064 \cdot 10^{-2}$
2^{-2}	$5.1947 \cdot 10^{-2}$
2^{-3}	$2.7449 \cdot 10^{-2}$
2^{-4}	$1.4134 \cdot 10^{-2}$
2^{-5}	$7.1748 \cdot 10^{-3}$
2^{-6}	$3.6152 \cdot 10^{-3}$
2^{-7}	$1.8146 \cdot 10^{-3}$

Table 1: Errors in the solutions computed by the logarithmic barrier method (see Example 2.1).

Later I will prove that, under certain conditions, $B(\cdot;\mu)$ has a unique minimizer x^*_μ in a neighborhood of x^* and that $x^*_\mu \to x^*$ as $\mu \to 0$.

For now, however, I will assume that the above method works, first applying it to Example 1.1 and then using it to deduce the optimality conditions for NLP (1–2).

Example 2.1 (Example 1.1, continued) *I apply the logarithmic barrier method to*

$$\begin{array}{ll} \min & f(x) \\ s.t. & h(x) \geq 0, \end{array}$$

where f and h are defined as in Example 1.1. I applied Newton's method to minimize $B(\cdot;\mu)$ for $\mu = 2^0, 2^{-1}, \ldots, 2^{-7}$, beginning with $x^{(0)} = (0.5, 0.5)$. The results are graphed in Figure 2, and the errors in the computed solutions are displayed in Table 1. The reader will notice that, at least for this example, $\|x^* - x^*_\mu\| = O(\mu)$ appears to hold.

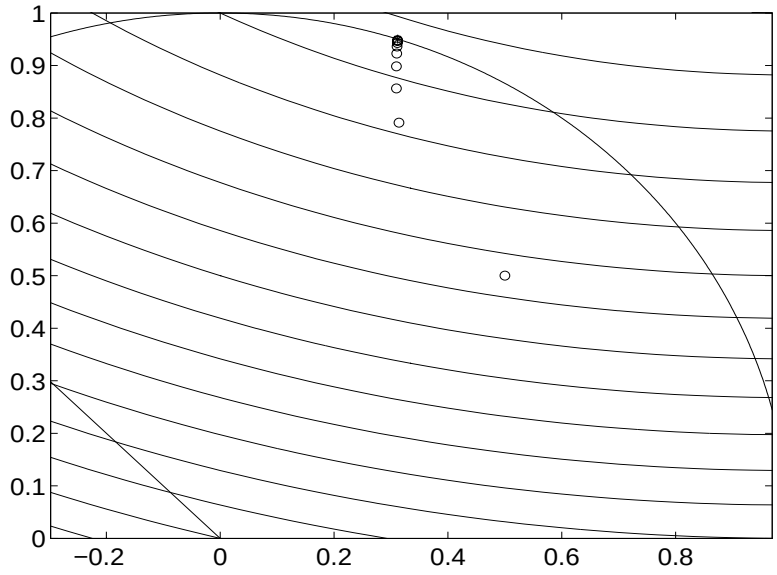

Figure 2: The approximate solutions to Example 1.1 computed by the logarithmic barrier method (see Example 2.1).

3 Optimality conditions for inequality-constrained problems: Introduction

I now assume that x^* is a local minimizer of (1–2) and that the logarithmic barrier method works as intended:

1. For all μ sufficiently small, $B(\cdot;\mu)$ has a unique local minimizer x^*_μ in a neighborhood of x^*; and
2. $x^*_\mu \to x^*$ as $\mu \to 0$.

Then, for all μ sufficiently small,

$$\nabla B(x^*_\mu;\mu) = 0.$$

I use the following notation: For any vector v, v^{-1} is the vector of the same size defined by $v_i^{-1} = 1/v_i$. (Of course, this is only defined if every component of v is nonzero.) Then it is easy to show that

$$\begin{aligned} \nabla B(x;\mu) &= \nabla f(x) - \mu \sum_{i=1}^{p} \frac{1}{g_i(x)} \nabla g_i(x) \\ &= \nabla f(x) - \nabla g(x)(\mu g(x)^{-1}). \end{aligned}$$

In particular, $\nabla B(x^*_\mu;\mu) = 0$ implies that

$$\nabla f(x^*_\mu) = \nabla g(x^*_\mu)(\mu g(x^*_\mu)^{-1}).$$

Since $x^*_\mu \to x^*$ as $\mu \to 0$, this suggests that there exists $\lambda^* \in \mathrm{R}^p$ with

$$\mu g(x^*_\mu)^{-1} \to \lambda^* \text{ as } \mu \to 0$$

and

$$\nabla f(x^*) = \nabla g(x^*)\lambda^*.$$

I will now assume that such a λ^* exists (rigorous proofs will be given later). Then, since $g(x^*_\mu) > 0$ for all $\mu > 0$, it follows that

$$\lambda^* \geq 0.$$

Moreover, for any i such that $g_i(x^*) > 0$,

$$\lambda^*_i = \lim_{\mu \to 0} \frac{\mu}{g_i(x^*_\mu)} = \frac{0}{g_i(x^*)} = 0.$$

This analysis suggests the following: If x^* is a local minimizer of (1–2), then there exists a Lagrange multiplier $\lambda^* \in \mathrm{R}^p$ such that

$$\nabla f(x^*) = \nabla g(x^*)\lambda^*.$$

In addition, λ^* satisfies the following properties:

$$\begin{aligned} &\lambda^*_i \geq 0 \text{ for all } i = 1, 2, \ldots, p, \\ &\quad g_i(x^*) > 0 \;\Rightarrow\; \lambda^*_i = 0. \end{aligned}$$

The second condition means that, for each i, either $g_i(x^*) = 0$ or $\lambda^*_i = 0$ (or both). This is referred to as the *complementarity* condition and can be written as

$$\lambda^*_i g_i(x^*) = 0, \text{ for all } i = 1, 2, \ldots, p.$$

If $g_i(x^*) = 0$, then constraint i is said to be *active* (or *binding*) at x^*. On the other hand, if $g_i(x^*) > 0$, then constraint i is *inactive*. The equation

$$\nabla f(x^*) = \nabla g(x^*)\lambda^* = \sum_{i=1}^{p} \lambda_i^* \nabla g_i(x^*),$$

together with the fact that $\lambda_i^* = 0$ if constraint i is inactive, means that $\nabla f(x^*)$ can be expressed as a linear combination of the gradients of the active constraints. Therefore, the inactive constraints do not play a role in the first-order necessary conditions. This is not surprising, as I pointed out above that a constraint that is inactive at the solution is locally unimportant.

My analysis suggests that the first-order necessary conditions for x^* to be a local minimizer of (1–2) are:

> There exists $\lambda^* \in \mathrm{R}^p$ that, together with x^*, satisfies
>
> $\nabla f(x^*) = \nabla g(x^*)\lambda^*$;
>
> $g(x^*) \geq 0$;
>
> $\lambda^* \geq 0$;
>
> $\lambda_i^* g_i(x^*) = 0$ for all $i = 1, 2, \ldots, p$.

Based on the theory for equality-constrained NLPs, the reader should not be surprised to learn that a constraint qualification is required to ensure that the Lagrange multiplier λ^* exists.

4 Another barrier function

Although the logarithmic barrier function is the most popular barrier function, it is not the only one. An alternative is

$$\tilde{B}(x;\mu) = f(x) + \mu \sum_{i=1}^{p} \frac{1}{g_i(x)}.$$

Assuming that x is strictly feasible, it is clear that $B(x;\mu) \to \infty$ as x approaches the boundary of the feasible set.

Much the same theory can be developed for the barrier function $\tilde{B}$ as for the logarithmic barrier function B. However, B is usually preferred in practice because the logarithm increases so slowly as the boundary is approached.