

Infeasible point methods for inequality-constrained nonlinear programming (unfinished)

Mark S. Gockenbach

1 Introduction

One potential drawback of the logarithmic barrier method is that it is a feasible point method (that is, the method only considers feasible points). In some cases, this might be considered an advantage of the method, since every estimate generated by the algorithm is feasible. However, a feasible point is required to start the method, and for certain problems it may be difficult to find such a point.

I will now present two methods that do not require feasible starting points.

2 Transforming the inequality constraints to equations

The inequality

$$h_i(x) \geq 0$$

is equivalent to the equation

$$\min\{h_i(x), 0\}^2 = 0.$$

The reason for writing $\min\{h_i(x), 0\}^2 = 0$ instead of the equivalent $\min\{h_i(x), 0\} = 0$ is that the function

$$x \mapsto \min\{h_i(x), 0\}^2 \tag{1}$$

is continuously differentiable everywhere, assuming h_i itself is continuously differentiable. The same fails to be true for

$$x \mapsto \min\{h_i(x), 0\}$$

for most functions h_i . The reader should notice, however, that (1) is typically not twice continuously differentiable (even if h_i itself is); the Hessian usually has a discontinuity at point x where $h_i(x) = 0$. I will discuss this further below.

When h is vector-valued, I will write $\max\{h(x), 0\}^2$ for the vector whose i th component is $\max\{h_i(x), 0\}^2$. Defining

$$g(x) = \min\{h(x), 0\}^2,$$

the NLP

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

can be equivalently formulated as

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0, \end{aligned}$$

and then solved by any method for equality-constrained NLPs, such as the quadratic penalty method or the augmented Lagrangian method. I will assume throughout the following discussion that h is at least continuously differentiable. The gradient of g is defined by the following formulas:

$$\nabla g_i(x) = \begin{cases} 2h_i(x)\nabla h_i(x), & h_i(x) < 0, \\ 0, & h_i(x) > 0. \end{cases} \quad (2)$$

Since

$$2h_i(x)\nabla h_i(x) \rightarrow 0 \text{ as } h_i(x) \rightarrow 0,$$

it follows that ∇g_i can be extended to a continuous function, and hence, by some further reasoning, that g is continuously differentiable. If h is twice differentiable, then

$$\nabla^2 g_i(x) = \begin{cases} 2h(x)\nabla^2 h_i(x) + 2\nabla h_i(x)\nabla h_i(x)^T, & h_i(x) < 0, \\ 0, & h_i(x) > 0. \end{cases}$$

If $h_i(x^*) = 0$, $h_i(x) < 0$, and $x \rightarrow x^*$, then

$$\nabla^2 g_i(x) \rightarrow 2\nabla h_i(x^*)\nabla h_i(x^*)^T,$$

and there is no reason to assume that this limit is the zero matrix. Therefore, $\nabla^2 g_i$ typically has a singularity at points x^* satisfying $h_i(x^*) = 0$. Moreover, if $i \in \mathcal{A}(x^*)$, then $\nabla g_i(x^*) = 0$, which means that x^* cannot be a regular point if any constraint $h_i(x) \geq 0$ is active at x^* . Thus, if x^* is a local minimizer of the original NLP and $\mathcal{A}(x^*) \neq \emptyset$, then the transformed NLP suffers from both a lack of smoothness and a lack of regularity at x^* . This means that much of the theory derived earlier does not hold, and there is no reason to expect good behavior from algorithms such as the augmented Lagrangian method.

Example 2.1 *Using the above method, I solved*

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

where $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ and $h : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ are defined by

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

I used a quasi-Newton method to solve the unconstrained minimization problems, and the stopping test was

$$\|g(x)\| \leq 10^{-8},$$

where, as above, $g(x) = \min\{h(x), 0\}^2$. The initial Lagrange multiplier for the augmented Lagrangian method was the zero vector, and the initial value of μ was 1.0, with μ reduced by a factor of 10 at each iteration. The algorithm took 21 iterations to converge, and

$$h_1(x^{(21)}) \doteq -7.1907 \cdot 10^{-5}.$$

The first constraint is the only active constraint at the solution x^* , so $h_1(x^{(21)}) \doteq 0$ should hold. The reader should notice that $h_1(x^{(21)})^2 \leq 10^{-8}$ is satisfied.

By way of comparison, I solved the same problem as an equality-constrained NLP, using the fact that only the first constraint is active at the solution:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & h_1(x) = 0. \end{aligned}$$

I used the augmented Lagrangian method with the same parameters and initial Lagrange multiplier estimate. The algorithm required only 6 iterations, and

$$h_1(x^{(6)}) \doteq -9.7915 \cdot 10^{-9}.$$

These results demonstrate the effects of the singularity in g at x^* .

3 Adding slack variables

Another approach transforms the inequality constraint $h(x) \geq 0$ to an equation by subtracting a vector s of *slack variables*:

$$h(x) \geq 0 \Rightarrow h(x) - s = 0, \quad s \geq 0.$$

The NLP

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

can equivalently be written as

$$\begin{aligned} \min \quad & f(x) & (3) \\ \text{s.t.} \quad & h(x) - s = 0 & (4) \\ & s \geq 0, & (5) \end{aligned}$$

where now the independent variables are x, s .

The augmented Lagrangian function for (3-5) is

$$L(x, s; \lambda; \mu) = f(x) - \lambda \cdot (h(x) - s) + \frac{1}{2\mu} \|h(x) - s\|^2,$$

and it is required to minimize $L(\cdot, \cdot; \lambda; \mu)$ subject to the constraint that $s \geq 0$. It would appear, then, that I have just traded one inequality-constrained problem for another (and the new problem has more variables!). However,

$$\min \{L(x, s; \lambda; \mu) : x \in \mathbb{R}^n, s \in \mathbb{R}^p, s \geq 0\} = \min \{ \min \{L(x, s; \lambda; \mu) : s \in \mathbb{R}^p, s \geq 0\} : x \in \mathbb{R}^n \},$$

and the inner minimization

$$\min \{L(x, s; \lambda; \mu) : s \in \mathbb{R}^p, s \geq 0\}$$

can be done analytically, eliminating s from the problem. In fact,

$$\begin{aligned} L(x, s; \lambda; \mu) &= f(x) - \sum_{i=1}^p \lambda_i (h_i(x) - s_i) + \frac{1}{2\mu} \sum_{i=1}^p (h_i(x) - s_i)^2 \\ &= f(x) + \sum_{i=1}^p \phi_i(s_i), \end{aligned}$$

where

$$\phi_i(s_i) = \frac{1}{2\mu} (h_i(x) - s_i)^2 - \lambda_i (h_i(x) - s_i).$$

Therefore, the problem of minimizing $L(x, s; \lambda; \mu)$ with respect to $s \geq 0$ reduces to the p (decoupled) problems of minimizing $\phi_i(s_i)$ with respect to $s_i \geq 0$, $i = 1, 2, \dots, p$. The function ϕ_i is a quadratic function of a scalar variable, and the (unconstrained) minimizer is

$$s_i = h_i(x) - \mu\lambda_i.$$

If this value of s_i is negative, then clearly $s_i = 0$ minimizes ϕ_i over the interval $[0, \infty)$. Therefore, the optimal value of s_i is given by

$$s_i^* = \begin{cases} h_i(x) - \mu\lambda_i, & h_i(x) > \mu\lambda_i, \\ 0, & h_i(x) \leq \mu\lambda_i, \end{cases}$$

and a little calculation shows that

$$\phi_i(s_i^*) = \begin{cases} \frac{1}{2\mu}h_i(x)^2 - \lambda_i h_i(x), & h_i(x) \leq \mu\lambda_i, \\ -\frac{1}{2}\mu\lambda_i^2, & h_i(x) > \mu\lambda_i. \end{cases}$$

I will define $\Phi : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ by

$$\Phi(\alpha; \beta; \gamma) = \begin{cases} \frac{1}{2\gamma}\alpha^2 - \beta\alpha, & \alpha \leq \beta\gamma, \\ -\frac{1}{2}\gamma\beta^2, & \alpha > \beta\gamma. \end{cases}$$

Then

$$\tilde{L}(x; \lambda; \mu) = \min \{L(x, s; \lambda; \mu) : s \in \mathbb{R}^p, s \geq 0\} = f(x) + \sum_{i=1}^p \Phi(h_i(x); \lambda_i; \mu),$$

and the augmented Lagrangian method will proceed by minimizing \tilde{L} , updating λ and possibly μ , and repeating until convergence.

To determine how to update the Lagrange multiplier, it is necessary to compute the gradient of \tilde{L} . First of all, it is easy to show that Φ is continuously differentiable with respect to α , and, for $\gamma > 0$,

$$\begin{aligned} \frac{\partial \Phi}{\partial \alpha}(\alpha, \beta, \gamma) &= \begin{cases} \frac{\alpha}{\gamma} - \beta, & \alpha \leq \beta\gamma, \\ 0, & \alpha > \beta\gamma \end{cases} \\ &= \begin{cases} -\left(\beta - \frac{\alpha}{\gamma}\right), & \beta - \frac{\alpha}{\gamma}, \\ 0, & \beta - \frac{\alpha}{\gamma} < 0 \end{cases} \\ &= -\max \left\{ \beta - \frac{\alpha}{\gamma}, 0 \right\}. \end{aligned}$$

Therefore,

$$\begin{aligned} \nabla_x [\Phi(h_i(x), \lambda_i, \mu)] &= \frac{\partial \Phi}{\partial \alpha}(h_i(x), \lambda_i, \mu) \nabla h_i(x) \\ &= -\max \{ \lambda_i - \mu^{-1}h_i(x), 0 \} \nabla h_i(x), \\ \nabla \tilde{L}(x; \lambda; \mu) &= \nabla f(x) - \nabla h(x) \max \{ \lambda - \mu^{-1}h(x), 0 \}, \end{aligned}$$

and hence

$$\nabla \tilde{L}(x; \lambda; \mu) = 0 \Rightarrow \nabla f(x) = \nabla h(x) \max \{ \lambda - \mu^{-1}h(x), 0 \}.$$

This formula suggests that λ be updated by the formula

$$\lambda \leftarrow \max \{ \lambda - \mu^{-1}h(x), 0 \}.$$

The reader should notice that this formula maintains nonnegativity for the Lagrange multiplier estimate λ .

The Hessian of $\tilde{L}(\cdot; \lambda; \mu)$ is derived as follows:

$$\begin{aligned} \nabla_{xx}^2 [\Phi(h_i(x), \lambda_i, \mu)] &= \begin{cases} \left(\frac{h_i(x)}{\mu} - \lambda_i\right) \nabla^2 h_i(x) + \frac{1}{\mu} \nabla h_i(x) \nabla h_i(x)^T, & h_i(x) < \mu\lambda_i, \\ 0, & h_i(x) > \mu\lambda_i, \end{cases} \\ \nabla^2 \tilde{L}(x; \lambda; \mu) &= \nabla^2 f(x) + \sum_{i=1}^p \nabla_{xx}^2 [\Phi(h_i(x), \lambda_i, \mu)]. \end{aligned}$$

As these formulas show, $\tilde{L}(\cdot; \lambda; \mu)$ is not twice continuously differentiable; indeed, $\nabla^2 \tilde{L}(\cdot; \lambda; \mu)$ has a singularity at values of x such that $h_i(x) = \mu\lambda_i$ for some i . However, unlike in the method

presented in Section 2, this lack of smoothness is not necessarily very harmful to the performance of the algorithm, since it usually does not occur at the solution. Indeed, if strict complementarity holds at the solution, μ is bounded away from 0, and (x, λ) is close to (x^*, λ^*) , then

$$\begin{aligned}\mu\lambda_i &> h_i(x) \doteq 0, \quad i \in \mathcal{A}(x^*), \\ h_i(x) &> \mu\lambda_i \doteq 0, \quad i \notin \mathcal{A}(x^*).\end{aligned}$$

Thus it is impossible that $h_i(x) = \mu\lambda_i$ for (x, λ) sufficiently close to (x^*, λ^*) .

Before turning to the analysis of this method, I present the following example.

Example 3.1 *I solved the NLP from Example 2.1 using the method described in this section. I used a quasi-Newton method to solve the unconstrained minimization problems, and the stopping test was*

$$\max \{\lambda_i h_i(x)\} \leq 10^{-8}.$$

The initial Lagrange multiplier for the augmented Lagrangian method was the zero vector, and the initial value of μ was 1.0, with μ reduced by a factor of 10 at each iteration. The algorithm took 6 iterations to converge, and

$$h_1(x^{(6)}) \doteq -9.7919 \cdot 10^{-9}.$$