

Lecture XX: Nonlinear Constraints – Barrier Function and Projected Augmented Lagrangian

Charles B. Moss¹

¹University of Florida

October 7, 2012

- 1 Algorithm DP (Model algorithm with a differentiable penalty function)
- 2 Barrier Function
- 3 Projected Augmented Lagrangian

Algorithm DP (Model algorithm with a differentiable penalty function)

- DP1** *Check termination conditions.* If x_t satisfies the optimality conditions, terminate with the current solution.
- DP2** *Minimize the penalty function.* Using x_t as the starting point, execute an algorithm to solve the unconstrained subproblem

$$\min_x P_l(x, p_t) \quad (1)$$

subject to safeguard conditions to protect against unboundedness. Let x_{t+1} be the estimate of this subproblem.

- DP3** *Increase the penalty parameter.* Set $t+1$ to a larger value than k . and go back to step DP1.

Consider solving the problem from my Agricultural Finance book

$$\begin{aligned} \max_{x,y} U(x,y) &= x^{0.60} y^{0.40} \\ \text{s.t. } 10,000 &\geq x^2 + \frac{y^2}{4} \end{aligned} \tag{2}$$

Barrier Function

- The barrier function works well when the constraint can be evaluated either above or below its constrained value.
 - Specifically, if the nonlinear constraint is an equality, then a sufficient penalty parameter can be derived to force that the constraint be close to the right hand side.
 - However, in certain cases, you may want to guarantee feasibility as a minimum condition. For this type of problem the barrier method may be preferred.

- Focusing on the logarithmic barrier function, we create a subproblem similar to that generated in the penalty parameter framework. Specifically

$$B(x, r) = f(x) - r \sum_{i=1}^m \ln(c_i(x)) \quad (3)$$

where r is the barrier parameter and the $\ln(\cdot)$ term in the maximization problem guarantees that the $c_i(x)$ will be greater than zero.

Projected Augmented Lagrangian

- Minos 5.5 uses the projected augmented Lagrangian algorithm to optimize problems involving nonlinear constraints.
- This algorithm as instituted in Minos solves a sequence of subproblems.
 - Each subproblem, or major iteration, solves a linearly constrained minimization (maximization) problem.
 - The constraints for this linear subproblem are the linear constraints plus the linearized nonlinear constraints.
 - Then like the straightforward penalty parameter method, the penalty can be increased to approach the nonlinear constraints with an arbitrary level of precision.

- Mathematically, the general problem can be written

$$\begin{aligned}
 \min_x \quad & f(x) + c'x + d'y - \lambda'(G(x) - G(x_t)) \\
 & \frac{1}{2}\rho(G(x) - G(x_t))'(G(x) - G(x_t)) \\
 \text{s.t.} \quad & G(x) + A_1y = b_1 \\
 & A_2x + A_3y = b_2 \\
 & l \leq \begin{pmatrix} x \\ y \end{pmatrix} \leq u
 \end{aligned} \tag{4}$$

In this problem, the major iteration involves the linearization of the constraints around the point x_t .

- The first step is to linearize the nonlinear constraints around the starting point of the major iteration

$$\begin{aligned} G(x) &\approx G(x_t) + \nabla_x G(x_t)(x - x_t) \\ &= G(x_t) + \nabla_x G(x_t)x - \nabla_x G(x_t)x_t. \end{aligned} \tag{5}$$

- Given this relationship, the linear constraints can then be written as

$$\begin{pmatrix} \nabla_x G(x_t) & A_1 \\ A_2 & A_3 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} b_1 + \nabla_x G(x_t)x_t - G(x_t) \\ b_2 \end{pmatrix} \tag{6}$$

Ignoring the slackness conditions.