Numerical Methods in Economics

MIT Press, 1998

Notes for Lecture 6: Constrained Optimization

March 4, 2020

Optimization Problems

- Canonical problem:

$$\begin{aligned} \min_x f(x) \\ s.t.\ \ g(x) = 0, \\ h(x) \le 0, \end{aligned}$$

  - $f : \mathbb{R}^n \to \mathbb{R}$ is the *objective function*
  - $g : \mathbb{R}^n \to \mathbb{R}^m$ is the vector of m *equality constraints*
  - $h : \mathbb{R}^n \to \mathbb{R}^\ell$ is the vector of ℓ *inequality constraints*.

- Examples:

  - Maximization of consumer utility subject to a budget constraint
  - Optimal incentive contracts
  - Portfolio optimization
  - Life-cycle consumption

- Assumptions

  - Always assume f, g, and h are continuous
  - Usually assume f, g, and h are C^1
  - Often assume f, g, and h are C^3

Linear Programming

- Canonical linear programming problem is

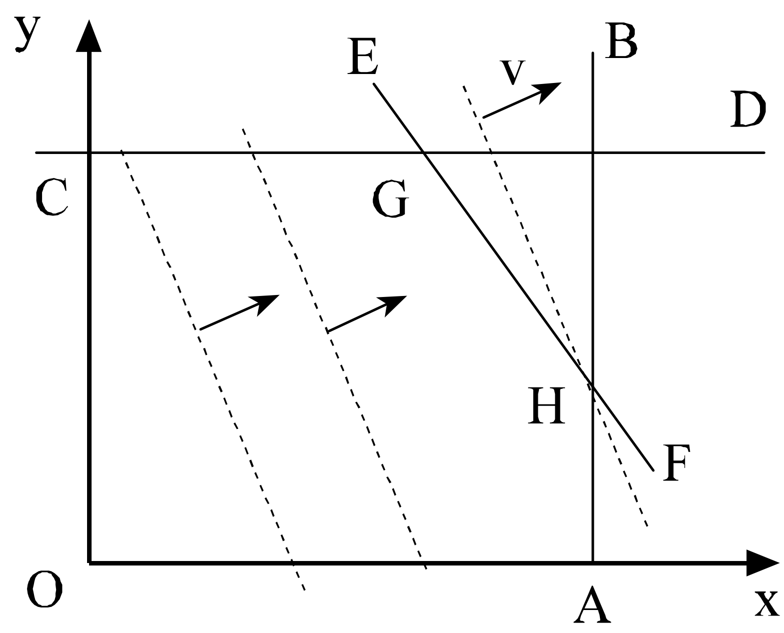
$$\begin{aligned} &\min_x a^\top x \\ &s.t. Cx = b, \\ &\quad x \geq 0. \end{aligned} \tag{1}$$

  - $Dx \leq f$: use *slack variables*, s, and constraints $Dx + s = f, s \geq 0$.
  - $Dx \geq f$: use $Dx - s = f, s \geq 0$, s is vector of *surplus variables*.
  - $x \geq d$: define $y = x - d$ and min over y
  - x_i free: define $x_i = y_i - z_i$, add constraints $y_i, z_i \geq 0$, and min over (y_i, z_i).

- Basic method is the *simplex method.* Figure 4.4 shows example:

$$\begin{aligned} \min_{x,y} \; & -2x - y \\ s.t. \; & x + y \leq 4, \quad x, y \geq 0, \\ & x \leq 3, \quad\; y \leq 2. \end{aligned}$$

  - Find some point on boundary of constraints, such as A.
  - Step 1: Note which constraints are active at A and points nearby.
  - Find feasible directions and choose steepest descent direction.
  - Figure 4.4 has two directions: from A: to B and to O, with B better.
  - Follow that direction to next vertex on boundary, and go back to step 1.
  - Continue until no direction reduces the objective: point H.
  - Stops in finite time since there are only a finite set of vertices.


y
E
B
v
D
C
G
H
F
O
A
x

- General History
  - Goes back to Dantzig (1951). (The real *Good Will Hunting*.)
  - Worst case time is exponential in number of variables and constraints
  - Fast on average – time is degree four polynomial in problem size
  - Software implementations vary in numerical stability
- Best software: CPLEX and GUROBI

Constrained Nonlinear Optimization

General problem:

$$\begin{aligned} &\min_x f(x) \\ s.t.\ &g(x) = 0 \\ &h(x) \leq 0 \end{aligned} \tag{4.7.1}$$

- $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$: objective function with n choices
  - $g : X \subseteq \mathbb{R}^n \to \mathbb{R}^m$: m equality constraints
  - $h : X \subseteq \mathbb{R}^n \to \mathbb{R}^\ell$: ℓ inequality constraints
  - $f, g,$ and h are C^2 on X
- Linear Independence Constraint Qualification (LICQ):
  - The set of constraints that hold with equality at a feasible point $x \in X$ is called the active set $A(x)$. Formally,

    $$A(x) = \{i \in I \mid g_i(x) = 0\} \cup E.$$

  - The linear independence constraint qualification (LICQ) holds at a point $x \in X$ if the gradients of all active constraints are linearly independent.

- Karush-Kuhn-Tucker (KKT) theorem: if there is a local minimum at x^* then there are multipliers $\lambda^* \in \mathbb{R}^m$ and $\mu^* \in \mathbb{R}^\ell$ such that x^* is a *stationary*, or *critical,* point of $\mathcal{L}$, the *Lagrangian*,
$$\mathcal{L}(x, \lambda, \mu) = f(x) + \lambda^\top g(x) + \mu^\top h(x) \tag{4.7.2}$$

- First-order conditions, $\mathcal{L}_x(x^*, \lambda^*, \mu^*) = 0$, imply that (λ^*, μ^*, x^*) solves
$$\begin{aligned} f_x + \lambda^\top g_x + \mu^\top h_x &= 0 \\ \mu_i h^i(x) &= 0\,, \quad i = 1, \cdots, \ell \\ g(x) &= 0 \\ h(x) &\leq 0 \\ \mu &\geq 0 \end{aligned} \tag{4.7.3}$$

- If LICQ holds then the multipliers are unique; otherwise, they are called “unbounded”.

- The KKT conditions are

$$\begin{aligned}
\nabla_x L(x^*, \lambda^*) = 0 \text{ i.e.} \nabla f(x^*) &= \sum_{i \in E \cup I} \lambda_i^* \nabla g_i(x^*) \\
g_i(x^*) &= 0, \forall i \in E \\
g_i(x^*) &\geq 0, \forall i \in I \\
\lambda_i^* &\geq 0, \forall i \in I \\
\lambda_i^* g_i(x^*) &= 0, \forall i \in E \cup I
\end{aligned}$$

- At a solution, x, all equality constraints must hold.
- Some inequality constraints will be active, that is, equal zero. For each solution x, define the active set of constraints

$$A(x) = E \cup \{i \in I | g_i(x) = 0\}$$

- Given x^* and $A(x^*)$, we say that the linear independence constraint qualification (LICQ) holds if the set of active constraint gradients $\{\nabla g_i(x^*) | i \in A(x^*)\}$ is linearly independent.

A Kuhn-Tucker Approach

- Idea: try all possible Kuhn-Tucker systems and pick best
  - Let $\mathcal{J}$ be the set $\{1, 2, \cdots, \ell\}$.
  - For a subset $\mathcal{P} \subset \mathcal{J}$, define the $\mathcal{P}$ problem, corresponding to a combination of binding and nonbinding inequality constraints

$$\begin{aligned} g(x) &= 0 \\ h^i(x) &= 0\,, \quad i \in \mathcal{P}, \\ \mu^i &= 0\,, \quad i \in \mathcal{J} - \mathcal{P}\ , \\ f_x + \lambda^\top g_x + \mu^\top h_x &= 0. \end{aligned} \tag{4.7.4}$$

  - Solve (or attempt to do so) each $\mathcal{P}$-problem
  - Choose the best solution among those $\mathcal{P}$-problems with solutions consistent with all constraints.

- We can do better in general.

Penalty Function Approach

- Many constrained optimization methods use a *penalty function* approach:
  - Replace constrained problem with related unconstrained problem.
  - Permit anything, but make it "painful" to violate constraints.
- Penalty function: for canonical problem

$$\begin{aligned} \min_x \quad & f(x) \\ s.t. \quad & g(x) = a, \\ & h(x) \leq b. \end{aligned} \tag{4.7.5}$$

construct the penalty function problem

$$\min_x \; f(x) + \frac{1}{2} P \left(\sum_i \left(g^i(x) - a_i \right)^2 + \sum_j \left(\max\left[0, h^j(x) - b_j \right] \right)^2 \right) \tag{4.7.6}$$

where $P > 0$ is the penalty parameter.

  - Denote the penalized objective in (4.7.6) $F(x; P, a, b)$.
  - Include a and b as parameters of $F(x; P, a, b)$.
  - If P is "infinite," then (4.7.5) and (4.7.6) are identical.
  - Hopefully, for large P, their solutions will be close.

- Problem: for large P, the Hessian of F, F_{xx}, is ill-conditioned at x away from the solution.

- Solution: solve a sequence of problems.

  – Solve $\min_x \ F(x; P_1, a, b)$ with a small choice of P_1 to get x^1.

  – Then execute the iteration

  $$x^{k+1} \in \arg\min_x \ F(x; P_{k+1}, a, b) \tag{4.7.7}$$

  where we use x^k as initial guess in iteration $k+1$, and $F_{xx}(x^k; P_{k+1}, a, b)$ as the initial Hessian guess (which is hopefully not too ill-conditioned)

- Shadow prices in (4.7.5) and (4.7.7):

  – Shadow price of a_i in (4.7.6) is $F_{a_i} = P(g^i(x) - a_i)$.

  – Shadow price of b_j in (4.7.6) is F_{b_j}; $P(h^j(x) - b_j)$ if binding, 0 otherwise.

- Theorem: Penalty method works with convergence of x and shadow prices as P_k diverges (under mild conditions)

- Simple example

  - Consumer buys good y (price is 1) and good z (price is 2) with income 5.
  - Utility is $u(y,z) = \sqrt{yz}$.
  - Optimal consumption problem is

$$\begin{aligned} &\max_{y,z} \sqrt{yz} \\ &s.t. \;\; y + 2z \leq 5. \end{aligned} \tag{4.7.8}$$

  with solution $(y^*, z^*) = (5/2, 5/4)$, $\lambda^* = 8^{-1/2}$.

  - Penalty function is

$$u(y,z) - \frac{1}{2} P(\max[0, y + 2z - 5])^2$$

  - Iterates are in Table 4.7 (stagnation due to finite precision)

Table 4.7

Penalty function method applied to (4.7.8)

k	P_k	$(y,z) - (y^*, z^*)$	Constraint violation	λ error
0	10	(8.8(-3), .015)	1.0(−1)	−5.9(−3)
1	10^2	(8.8(−4), 1.5(−3))	1.0(−2)	−5.5(−4)
2	10^3	(5.5(−5), 1.7(−4))	1.0(−3)	2.1(−2)
3	10^4	(−2.5(−4), 1.7(−4))	1.0(−4)	1.7(−4)
4	10^5	(−2.8(−4), 1.7(−4))	1.0(−5)	2.3(−4)

Sequential Quadratic Programming Method

- Special methods are available when we have a quadratic objective and linear constraints

$$\begin{aligned} & \min_x (x-a)^\top A (x-a) \\ s.t. \quad & b(x-s) = 0 \\ & c(x-q) \le 0 \end{aligned}$$

- Extensions of linear programming

- Excellent software includes CPLEX and GUROBI

- Sequential Quadratic Programming Method
  - Solution is stationary point of Lagrangian

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \lambda^\top g(x) + \mu^\top h(x)$$

  - Suppose that the current guesses are (x^k, λ^k, μ^k).
  - Let step size s^{k+1} solve approximating quadratic problem

$$\begin{array}{ll} \min_s & \mathcal{L}_x(x^k, \lambda^k, \mu^k)(x^k - s) + (x^k - s)^\top \mathcal{L}_{xx}(x^k, \lambda^k, \mu^k)(x^k - s) \\ s.t. & g(x^k) + g_x(x^k)(x^k - s) = 0 \\ & h(x^k) + h_x(x^k)(x^k - s) \leq 0 \end{array}$$

  - The next iterate is $x^{k+1} = x^k + \phi s^{k+1}$ for some ϕ
    - Could use linesearch to choose ϕ
    - λ^k and μ^k are also updated but we do not describe the detail here.
  - Proceed through a sequence of quadratic problems.
  - SQP method inherits many properties of Newton's method
    - rapid local convergence
    - can use quasi-Newton to approximate Hessian.

Domain Problems

- Suppose $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$, $g : X \subseteq \mathbb{R}^n \to \mathbb{R}^m$, $h : X \subseteq \mathbb{R}^n \to \mathbb{R}^\ell$, and we want to solve

$$\begin{aligned} \min_x & \; f(x) \\ s.t. & \; g(x) = 0 \\ & \; h(x) \leq 0 \end{aligned} \tag{4.7.1}$$

- The penalty function approach produces an unconstrained problem

$$\max_{x \in \mathbb{R}^n} F(x; P, a, b)$$

- Problem: $F(x; P, a, b)$ may not be defined for all x.

- Example: Consumer demand problem

$$\begin{aligned} & \max_{y,z} u(y, z) \\ & s.t. \; p\, y + q\, z \leq I. \end{aligned}$$

  – Penalty method

$$\max_{y,z} \; u(y, z) - \frac{1}{2} P(\max[0, \; p\, y + q\, z - I])^2$$

- Problem: $u\left(y,z\right)$ will not be defined for all y and z, such as

$$\begin{aligned} u(y,z) &= \log y + \log z \\ u(y,z) &= y^{1/3}z^{1/4} \\ u(y,z) &= \left(y^{1/6}+z^{1/6}\right)^{7/2} \end{aligned}$$

- Penalty method may crash when computer tries to evaluate $u\left(y,z\right)$!

- Solutions

  - Strategy 1: Transform variables

    * If functions are defined only for $x_i > 0$, then reformulate in terms of $z_i = \log x_i$
    * For example, let $\widetilde{y} = \log y$, $\widetilde{z} = \log z$, and solve

$$\max_{\widetilde{y},\widetilde{z}} \; u(e^{\widetilde{y}}, e^{\widetilde{z}}) - \frac{1}{2}P(\max[0,\; p\; e^{\widetilde{y}} + q\; e^{\widetilde{z}} - I])^2$$

    * Problem: log transformation may not preserve shape; e.g., concave function of x may not be concave in $\log x$

  - Strategy 2: Alter objective and constraint functions so that they are defined everywhere (see discussion above)

  - Strategy 3: Express the domain where functions are defined in terms of inequality constraints that are enforced by the algorithm at every step.

    * E.g., if utility function is $\log(x) + \log(y)$, then add constraints $x \geq \delta, y \geq \delta$ for some very small $\delta > 0$ (use, for example, $\delta \approx 10^{-6}$; don't use $\delta = 0$ since roundoff error may still allow negative x or y)
    * In general, you can avoid domain problems if you express the domain in terms of linear constraints.
    * If the domain is defined by nonlinear functions, then create new variables that can describe the domain in linear terms.

Active Set Approach

- Problems:
  - Kuhn-Tucker approach has too many combinations to check
    * some choices of $\mathcal{P}$ may have no solution
    * there may be multiple local solutions to others.
  - Sequential quadratic method can be slow if there are too many constraints.
  - Penalty function methods are costly since all constraints are in (4.7.5), even if only a few bind at solution.

- Solution: refine K-T with a *good sequence* of subproblems, ignoring constraints that you think won't be active at the solution.

  - Let $\mathcal{J}$ be the set $\{1, 2, \cdots, \ell\}$
  - for $\mathcal{P} \subset \mathcal{J}$, define the $\mathcal{P}$ problem

$$\begin{array}{ll} \min_x f(x) & \\ s.t.\ g(x) = 0, & (\mathcal{P}) \\ \quad h^i(x) \leq 0, \quad i \in \mathcal{P}. & \end{array} \tag{4.7.10}$$

  - Choose an initial set of constraints, $\mathcal{P}$, and solve (4.7.10-$\mathcal{P}$)If that solution satisfies all constraints, then you are done.
  - Otherwise
    - Add constraints which are violated by most recent guess
    - Periodically drop constraints in $\mathcal{P}$ which fail to bind
    - Increase penalty parameters
    - Repeat

- The simplex method for linear programing is really an active set method.

Interior-Point methods

- Consider

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^\top x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

where $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and A is an $m \times n$ matrix.

- Karush-Kuhn-Tucker conditions for this optimization problem are as follows.

$$A^\top \lambda + s = c \tag{2}$$

$$Ax = b \tag{3}$$

$$x_i s_i = 0, \quad i = 1, 2, \ldots, n \tag{4}$$

$$x \geq 0 \tag{5}$$

$$s \geq 0 \tag{6}$$

- Interior-point methods solve a sequence of perturbed problems.
  - Consider the following perturbation of the KKT conditions.

$$\begin{aligned} A^\top \lambda + s &= c && (7)\\ Ax &= b && (8)\\ x_i s_i &= \mu, \quad i = 1, 2, \ldots, n && (9)\\ x &> 0 && (10)\\ s &> 0 && (11) \end{aligned}$$

  - The complementarity condition (4) is replaced by (9) for some positive scalar $\mu > 0$.
  - Assuming that a solution $(x^{(0)}, \lambda^{(0)}, s^{(0)})$ to this system is given for some initial value of $\mu^{(0)} > 0$, interior-point methods decrease the parameter μ and thereby generate a sequence of points $(x^{(k)}, \lambda^{(k)}, s^{(k)})$ that satisfy the non-negativity constraints on the variables strictly, $x^{(k)} > 0$ and $s^{(k)} > 0$.
  - As μ is decreased to zero, a point satisfying the original first-order conditions is reached.
  - The set of solutions to the perturbed system,

$$C = \{x(\mu), \lambda(\mu), s(\mu) \mid \mu > 0\}$$

    is called the central path.
- Implementations must handle many details

- It is often difficult to find a feasible starting point $(x^{(0)}, \lambda^{(0)}, s^{(0)})$ of the perturbed system.
- Good initial guesses generally do not work! IPOPT will use good initial guesses.
- We need to solve (7) – (9) in each iteration and maintain 10 and 11.
- Newton's method can be used but better is to use path-following to maintain the inequalities.

The Logarithmic Barrier Method

- Consider

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \geq 0 \quad i \in I \end{aligned}$$

- Combine the objective function and constraints to define a penalty function

$$P(x; \mu) = f(x) - \mu \sum_{i \in I} \ln g_i(x),$$

  - $\mu > 0$ is called the barrier parameter
  - $\sum_{i \in I} \ln g_i(x)$ is called a logarithmic barrier function.
  - Each $-\ln g_i(x)$ term tends to infinity as x approaches the boundary of $g_i(x) \geq 0$ from the interior of the feasible region.
  - As μ converges to zero, the optimal solution $x^*(\mu)$ path of $\min_{x \in \mathbb{R}^n} P(x; \mu)$ converges to the optimal solution of the original problem.

- First-order conditions are

$$\nabla_x P(x;\mu) = \nabla f(x) - \sum_{i \in I} \frac{\mu}{g_i(x)} \nabla g_i(x) = 0.$$

- Now define for all $i \in I$

$$\nu_i(\mu) := \frac{\mu}{g_i(x)} \ .$$

  - Note that since $\mu > 0$ by definition we have that $\nu_i(\mu) > 0$.
  - Thus, at a stationary point of the penalty function the following conditions hold.

$$\begin{aligned}
\nabla f(x) - \sum_{i \in I} \nu_i \nabla g_i(x) &= 0 \\
g_i(x) - s_i &= 0 \quad \text{for all } i \in I \\
\nu_i s_i &= \mu \quad \text{for all } i \in I \\
\nu_i &> 0 \quad \text{for all } i \in I \\
s_i &> 0 \quad \text{for all } i \in I
\end{aligned}$$