Constrained optimization

Kevin Carlberg (Sandia National Laboratories)

August 13, 2019

Theory

Outline

Constrained optimization

Theory, methods, and software for problems exihibiting the characteristics below

- Convexity:
  - convex: local solutions are global
  - non-convex: local solutions are not global
- Optimization-variable type:
  - continuous: gradients facilitate computing the solution
  - discrete: cannot compute gradients, NP-hard
- Constraints:
  - unconstrained: simpler algorithms
  - constrained: more complex algorithms; must consider feasibility
- Number of optimization variables:
  - low-dimensional: can solve even without gradients
  - high-dimensional: requires gradients to be solvable in practice

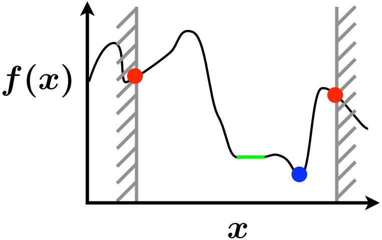
Constrained optimization

- This lecture considers constrained optimization

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & c_i(x) = 0, \quad i = 1, \ldots, n_e \\ & d_j(x) \geq 0, \quad j = 1, \ldots, n_i \end{aligned} \tag{1}$$

- Equality constraint functions: $c_i : \mathbf{R}^n \to \mathbf{R}$
- Inequality constraint functions: $d_j : \mathbf{R}^n \to \mathbf{R}$
- Feasible set: $\Omega = \{x \mid c_i(x) = 0,\, d_j(x) \geq 0, i = 1, \ldots, n_e,\, j = 1, \ldots, n_i\}$
- We assume all functions are twice-continuously differentiable

What is a solution?



- Global minimum: A point $x^* \in \Omega$ satisfying $f(x^*) \leq f(x)\ \forall x \in \Omega$
- Strong local minimum: A neighborhood $\mathcal{N}$ of $x^* \in \Omega$ exists such that $f(x^*) < f(x)$ $\forall x \in \mathcal{N} \cap \Omega$.
- Weak local minima A neighborhood $\mathcal{N}$ of $x^* \in \Omega$ exists such that $f(x^*) \leq f(x)$ $\forall x \in \mathcal{N} \cap \Omega$.

Convexity

As with the unconstrained case, conditions hold where any local minimum is the global minimum:

- $f(x)$ convex
- $c_i(x)$ affine ($c_i(x) = A_i x + b_i$) for $i = 1, \ldots, n_e$
- $d_j(x)$ convex for $j = 1, \ldots, n_i$

Active set

- The active set at a feasible point $x \in \Omega$ consists of the equality constraints and the inequality constraints for which $d_j(x) = 0$

$$\mathcal{A}(x) = \{c_i\}_{i=1}^{n_i} \cup \{d_j \mid d_j(x) = 0\}$$

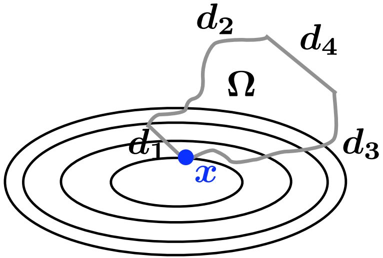


Figure 1: $\mathcal{A}(x) = \{d_1,\ d_3\}$

First-order necessary conditions

Words: the function cannot decrease by moving in feasible directions

Theorem (First-order necessary KKT conditions for local minima)

If x^* is a weak local minimum, then

$$\nabla f(x^*) - \sum_{i=1}^{n_e} \gamma_i \nabla c_i(x^*) - \sum_{j=1}^{n_i} \lambda_j \nabla d_j(x^*) = 0$$

$$\lambda_j \geq 0, \quad j = 1, \ldots, n_i$$

$$c_i(x^*) = 0, \quad i = 1, \ldots, n_e$$

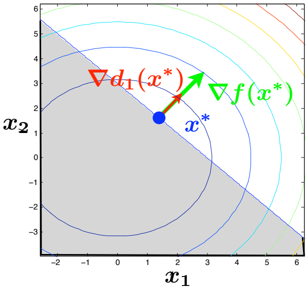
$$d_j(x^*) \geq 0, \quad j = 1, \ldots, n_i$$

$$\lambda_j d_j(x^*) = 0, \quad j = 1, \ldots, n_i$$

- Stationarity, Dual feasibility, Primal feasibility ($x^* \in \Omega$), Complementarity conditions, Lagrange multipliers γ_i, λ_j

Intuition for stationarity and dual feasibility

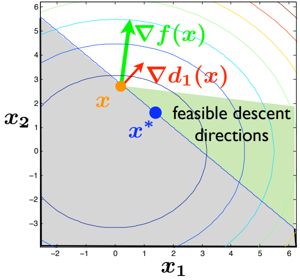
$$\begin{aligned} \underset{x \in \mathbf{R}^n}{\text{minimize}} \quad & f(x) = x_1^2 + x_2^2 \\ \text{subject to} \quad & d_1(x) = x_1 + x_2 - 3 \geq 0 \end{aligned}$$



- The solution is $x^* = (1.5, 1.5)$

Intuition for stationarity and dual feasibility (continued)

- The KKT conditions say $\nabla f(x^*) = \lambda_1 \nabla d_1(x^*)$ with $\lambda_1 \geq 0$
- Here, $\nabla f(x^*) = [3,3]^T$, while $\nabla d_1(x^*) = [1.5, 1.5]^T$, so these conditions are indeed verified with $\lambda_1 = 2 \geq 0$
- This is obvious from the figure: if $\nabla f(x^*)$ and $\nabla d_1(x^*)$ were "misaligned," there would be feasible descent directions!



- This gives us some intuition for stationarity and dual feasibility

Lagrangian

Definition (Lagrangian)

The Lagrangian for (1) is

$$\mathcal{L}(x,\gamma,\lambda) = f(x) - \sum_{i=1}^{n_e} \gamma_i c_i(x) - \sum_{j=1}^{n_i} \lambda_j d_j(x)$$

- Stationarity in the sense of KKT is equivalent to stationarity of the Lagrangian with respect to x:

$$\nabla_x \mathcal{L}(x,\gamma,\lambda) = \nabla f(x) - \sum_{i=1}^{n_e} \gamma_i \nabla c_i(x) - \sum_{j=1}^{n_i} \lambda_j \nabla d_j(x)$$

- KKT stationarity $\Leftrightarrow \nabla_x \mathcal{L}(x^*,\gamma,\lambda) = 0$

Lagrange multipliers

- **Lagrange multipliers** γ_i and λ_j arise in constrained minimization problems
- They tell us about the *sensitivity* of $f(x^*)$ to the constraints.
  - γ_i and λ_j indicate how hard f is "pushing" or "pulling" the solution against c_i and d_j.
- If we perturb the right-hand side of the ith equality constraint so that $c_i(x) \geq -\epsilon \|\nabla c_i(x^*)\|$, then

$$\frac{df(x^*(\epsilon))}{d\epsilon} = -\gamma_i \|\nabla c_i(x^*)\|.$$

- If we perturb the jth inequality so that $d_j(x) \geq -\epsilon \|\nabla d_j(x^*)\|$, then

$$\frac{df(x^*(\epsilon))}{d\epsilon} = -\lambda_j \|\nabla d_i(x^*)\|.$$

Intuition for complementarity

- We just saw that non-participating constraints have zero Lagrange multipliers
- The complementarity conditions are

$$\lambda_j d_j(x^*) = 0, \quad j = 1, \ldots, n_i$$

- This means that each inequality constraint must be either:
  1. Inactive (non-participating): $d_j(x^*) > 0$, $\lambda_j = 0$,
  2. Strongly active (participating): $d_j(x^*) = 0$ and $\lambda_j > 0$, *or*
  3. Weakly active (active but non-participating): $d_j(x^*) = 0$ and $\lambda_j = 0$

Second-order conditions for **unconstrained** problems

- Recall, second-order conditions for unconstrained problems

Theorem (Necessary conditions for a weak local minimum)

A1. $\nabla f(x^*) = 0$ *(stationary point)*
A2. $\nabla^2 f(x^*) \succeq 0$ *(*$p^T \nabla^2 f(x^*) p \geq 0$ *for all* $p \neq 0$*)*

Theorem (Sufficient conditions for a strong local minimum)

B1. $\nabla f(x^*) = 0$ *(stationary point)*
B2. $\nabla^2 f(x^*) \succ 0$ *(*$p^T \nabla^2 f(x^*) p > 0$ *for all* $p \neq 0$*).*

Second-order conditions for **constrained** problems

- We make an analogous statement for constrained problems, but limit the directions p to the critical cone $\mathcal{C}(x^*, \gamma)$
- Critical cone $\mathcal{C}(x^*, \gamma)$: set of directions that "adhere" to equality and active inequality constraits

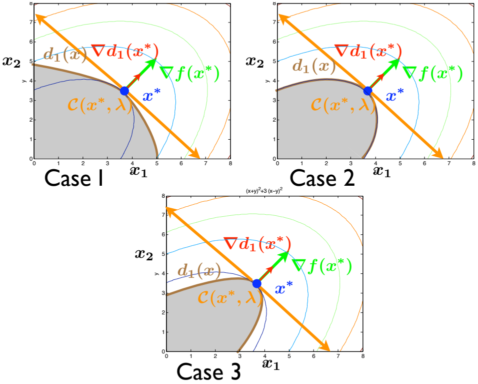
Theorem (Necessary conditions for a weak local minimum)

D1. KKT conditions hold
D2. $p^T \nabla_x^2 \mathcal{L}(x^*, \gamma) p \geq 0$ *for all* $p \in \mathcal{C}(x^*, \gamma)$

Theorem (Sufficient conditions for a strong local minimum)

E1. KKT conditions hold
E2. $p^T \nabla_x^2 \mathcal{L}(x^*, \gamma) p > 0$ *for all* $p \in \mathcal{C}(x^*, \gamma)$.

Intuition for second-order conditions



- **Case 1**: E1 and E2 are satisfied (sufficient conditions hold)
- **Case 2**: D1 and D2 are satisfied (necessary conditions hold)
- **Case 3**: D1 holds, D2 does not (necessary conditions **failed**)
  - Can reduce objective by curving around boundary!

Algorithms

Outline

Constrained optimization algorithms

- Linear programming (LP)
  - Simplex method: created by Dantzig in 1947. Birth of the modern era in optimization
  - Interior-point methods
- Nonlinear programming (NLP)
  - Penalty methods
  - Augmented Lagrangian methods
  - Interior-point methods
  - Sequential quadratic/convex programming methods
- Almost all of these methods rely on line-search and trust region methodologies from unconstrained optimization!
- Algorithmic approaches for constrained optimization
  1. Solve a sequence of unconstrained problems (penalty, interior-point)
  2. Solve a sequence of simpler problems (SQP, SCP)

Penalty methods

$$\text{minimize } f(x) \quad \text{subject to } c_i(x) = 0, \quad i = 1, \dots, n_i$$

- Penalty methods combine the objective and constraints
- Smooth penalty functions

$$\text{minimize } f(x) + \frac{\mu}{2} \sum_{i=1}^{n_i} c_i^2(x)$$

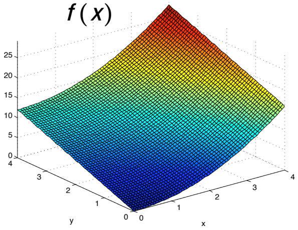
- Non-smooth penalty functions

$$\text{minimize } f(x) + \mu \sum_{i=1}^{n_i} |c_i(x)|$$

Penalty methods example (smooth)

- Original problem:

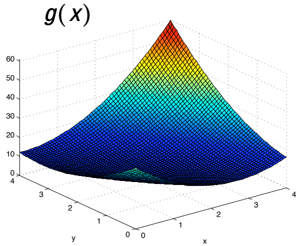
$$\text{minimize } f(x) = x_1^2 + 3x_2, \quad \text{subject to } x_1 + x_2 - 4 = 0$$



Penalty methods example (smooth)

- Penalty formulation:

$$\text{minimize } g(x) = x_1^2 + 3x_2 + \frac{\mu}{2}(x_1 + x_2 - 4)^2$$



- A valley is created along the constraint $x_1 + x_2 - 4 = 0$

Penalty methods tradeoffs

1. Smoothness v. exactness
   - *Smooth penalty*: preserve smoothness (easier to solve), but must solve a sequence of problems for increasing μ
   - *Non-smooth penalty*: it is exact (solve only one problem), but objective no longer smooth (harder to solve)
2. Size of penalty parameter
   - *Large*: function less likely to be unbounded below and closer to exact solution, but more ill-conditioned Hessians
   - *Small*: Better conditioned Hessians, but slower convergence

Interior-point methods

- These methods are also known as "barrier methods," because they build a barrier at the inequality constraint boundary

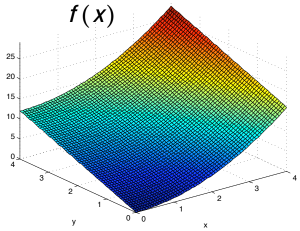
$$\begin{aligned} \text{minimize} \quad & f(x) - \mu \sum_{i=1}^{n_i} \log d_j(x) \\ \text{subject to} \quad & c_i(x) = 0, \quad i = 1, \ldots, n_e \end{aligned}$$

- **Slack variables**: s_i, indicates distance from constraint boundary
- Solve a sequence of problems with μ decreasing

Interior-point methods example

- Original problem:

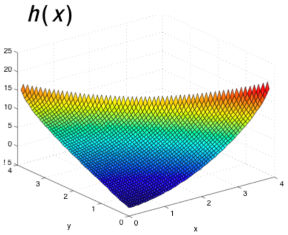
$$\text{minimize } f(x) = x_1^2 + 3x_2, \quad \text{subject to } -x_1 - x_2 + 4 \geq 0$$



Interior-points methods example

- Interior-point formulation:

$$\text{minimize } h(x) = x_1^2 + 3x_2 - \mu \log\left(-x_1 - x_2 + 4\right)$$



- A barrier is created along the boundary of the inequality constraint $x_1 + x_2 - 4 = 0$

Sequential quadratic programming

- Perhaps the most effective algorithm
- Solve a quadratic programming (QP) subproblem at each iteration

$$
\begin{aligned}
\text{minimize} \quad & \frac{1}{2} p^T \nabla^2_{xx} \mathcal{L}(x_k, \lambda_k) p + \nabla f(x_k)^T p \\
\text{subject to} \quad & \nabla c_i(x_k) T p + c_i(x_k) = 0, \quad i = 1, \ldots, n_e \\
& \nabla d_j(x_k)^T p + d_j(x_k) \geq 0, \quad j = 1, \ldots, n_i
\end{aligned}
$$

- When $n_i = 0$, this is equivalent to Newton's method on the KKT conditions
- When $n_i > 0$, this corresponds to an "active set" method, where we keep track of the set of active constraints $\mathcal{A}(x_k)$ at each iteration
- Sequential convex programming (SCP) is a variant wherein the subproblem is convex, but need not be quadratic

Summary

- Many concepts from the unconstrained case extend to the constrained case
  - First-order and second-order optimality
- To handle constraints, we make a few adjustments
  - Modify notions of first-order and second-order optimality
  - Introduce Lagrange multipliers to quantify the effect of constraints
- Algorithmic approaches for constrained optimization
  1. Solve a sequence of unconstrained problems (penalty, interior-point)
  2. Solve a sequence of simpler problems (SQP, SCP)