Advanced Mathematical Programming IE417

Lecture 22

Dr. Ted Ralphs

Reading for This Lecture

- Sections 10.1-10.2

Methods of Feasible Directions

- In Chapter 9, we looked at methods of using unconstrained optimization techniques on constrained problems.
- These methods enforced some of the constraints implicitly.
- Now, we look at the methods that explicitly enforce feasibility while ensuring convergence.
- Recall the concept of an improving, feasible direction.

Feasible and Improving Directions

Definition 1. *Let S be a nonempty set in $\mathbb{R}^n$ and let $x^* \in clS$. The* cone of feasible directions *of S at x^* is given by*

$$D = \{d : d \neq 0 \text{ and } x^* + \lambda d \in S, \forall \lambda \in (0, \delta), \exists \delta > 0\}$$

Definition 2. *Let S be a nonempty set in $\mathbb{R}^n$ and let $x^* \in clS$. Given a function $f : \mathbb{R}^n \to \mathbb{R}$, the cone of improving directions of f at x^* is given by*

$$F = \{d : f(x^* + \lambda d) < f(x^*) \forall \lambda \in (0, \delta), \exists \delta > 0\}$$

The Case of Linear Constraints

- Consider the problem

$$\begin{aligned} &\min f(x) \\ &\text{s.t. } Ax \le b \\ &\qquad Cx = d \end{aligned}$$

- The direction d is a feasible direction from the point x^* if and only if $A_1 d \le 0$ and $Cd = 0$ where A_1 are the constraints binding at x^*.
- If $\bigtriangledown f(x^*)^\top d < 0$, then d is an improving direction.

Generating Improving Feasible Directions (Linear Case)

- We want to generate a direction for which
  - $\nabla f(x^*)^\top d < 0$
  - $A_1 d \leq 0$
  - $Cd = 0$
- Idea: Solve the following optimization problem

$$\begin{aligned} \min\ & \nabla f(x^*)^\top d \\ \text{s.t.}\ & A_1 d && \leq 0 \\ & Cd && = 0 \end{aligned}$$

- We also need a normalizing constraint.

Remember Way Back When...

- Recall the derivation of the KKT conditions.
- This is exactly the system of equations from which the KKT conditions can be derived.
- Hence, x^* is a KKT point if and only if the optimal solution to this program is 0.
- Also, recall homework problem $\#4.29$.
- We can obtain a feasible solution to this program simply by projecting the direction $-\bigtriangledown f(x^*)$ onto the null space of $\begin{bmatrix} A_1 \\ C \end{bmatrix}$.

Review : Optimality Conditions

- If x^* is a local minimum, then $F \cap D = \emptyset$.
- The converse is not true.
- Given a feasible $x^* \in \mathbb{R}^n$, set $I = \{i : g_i(x^*) = 0\}$.
- Define $F_0 = \{d : \bigtriangledown f(x^*)^\top d < 0\}$ and ${F_0}' = \{d : d \neq 0, \bigtriangledown f(x^*)^\top d \leq 0\}$. Then $F_0 \subseteq F \subseteq {F_0}'$.
- Define $G_0 = \{d : \bigtriangledown g_i(x^*)^\top d < 0 \ \forall i \in I\}$ and ${G_0}' = \{d : d \neq 0, \bigtriangledown g_i(x^*)^\top d \leq 0 \ \forall i \in I\}$. Then $G_0 \subseteq D \subseteq {G_0}'$.

Review : Fritz-John Conditions

- If x^* is a local minimum, then $F_0 \cap G_0 = \emptyset$.
- $F_0 \cap G_0 = \emptyset$ if and only if there exists $\mu, v \in \mathbb{R}^m$ such that

$$\mu_0 \bigtriangledown f(x^*) + \sum \mu_i \bigtriangledown g_i(x^*) + \sum v_i \bigtriangledown h_i(x^*) = 0$$

$$\mu_i g_i(x^*) = 0 \; \forall i \in [1, m]$$

$$\mu \geq 0$$

$$(\mu, v) \neq 0$$

- These are the FJ conditions.

Review : KKT Conditions

- Assuming that $\bigtriangledown g_i(x^*)$ and $\bigtriangledown h_i(x^*)$ are linearly independent, then $\mu_0 > 0$ and we obtain the KKT conditions

$$\begin{aligned} \bigtriangledown f(x^*) + \sum \mu_i \bigtriangledown g_i(x^*) + \sum v_i \bigtriangledown h_i(x^*) &= 0 \\ \mu_i g_i(x^*) &= 0 \; \forall i \in [1, m] \\ \mu &\geq 0 \end{aligned}$$

- x^* is a KKT point if and only if $F_0 \cap {G_0}' = \emptyset$.

Method of Zoutendijk (Linear Case)

- Once we have found an improving, feasible direction we can do a line search in that direction.
- Recall homework problem #8.8.
- Since we have a convex feasible region, the line search problem reduces to:

$$
\begin{aligned}
\min f(x_k + \lambda d_k) & \\
0 \le \lambda \le \lambda_{\max} & \\
\lambda_{\max} = \min\{b_i{}'/d_i{}' : d_i{}' > 0\} & \\
b' = b_2 - A_2 x_2, d' = A_2 d_k &
\end{aligned}
$$

The Case of Nonlinear Constraints

- Recall that if f and g_i are differentiable and continuous, then if $\bigtriangledown f(x^*)^\top d < 0$ and $\bigtriangledown g_i(x^*)^\top d < 0 \ \forall i \in I$, then d is an improving, feasible direction from x^*.
- Consider the following optimization problem P:

$$\begin{aligned} &\min z \\ &\text{s.t} \quad \bigtriangledown f(x^*)^\top d \le z \\ &\qquad\;\; \bigtriangledown g_i(x^*)^\top d \le z \ \forall i \in I \end{aligned}$$

- The optimal solution to this problem is 0 if and only if x^* is an FJ point.

Method of Zoutendijk (Nonlinear Constraints)

- As before, we generate an improving, feasible direction d by solving a normalized version of the problem P.
- Perform a line search in direction d subject to feasibility constraints

$$\begin{aligned} &\min f(x_k + \lambda d_k) \\ &\text{s.t} \quad g_i(x_k + \lambda d_k) \leq 0 \, \forall i \end{aligned}$$

- Note the difference from what is in the book.

Convergence of Zoutendijk

- The direction-finding problem only uses the binding constraints.
- "Nearly binding" constraints can cause very short steps to be taken and also drastic changes in direction.
- This causes the algorithmic map not to be closed.
- This can cause "jamming" and slow convergence.
- Idea: Use constraints that are "nearly binding" in the direction-finding problem.
- Even this is not enough to guarantee convergence.

Method of Topkis and Veinott

- Try to eliminate drastic changes in direction by accounting for all constraints.
- Use the following direction-finding problem:

$$\begin{aligned} &\min z \\ &\text{s.t} \quad \nabla f(x^*)^\top d - z \leq 0 \\ &\qquad \nabla g_i(x^*)^\top d - z \leq -g_i(x^*) \\ &\qquad -1 \leq d_j \leq 1 \end{aligned}$$

- This is enough to guarantee convergence to an FJ point.

Convergence of Topkis and Veinott

- Note that the solution to the direction-finding problem is feasible and improving.
- Also, the optimal solution is 0 if and only if the current point is an FJ point.
- Taking all the constraints into account eliminates drastic changes in direction and ensures that the algorithmic map is closed.
- Under the assumption that all the functions involved are continuously differentiable, a sequence $\{x_k\}$ is generated by this algorithm, then all accumulation points are FJ points.

Comments on Topkis and Veinott

- Note that this is essentially a steepest descent algorithm.
- The steepest decent direction is found through first-order function approximations in a neighborhood of the current point.
- However, as with steepest descent, we may expect slow convergence and "zig-zagging" in some cases.

Successive Linear Programming

- Basic Approach: Make a first order linear programming approximation of the problem.
- Consider the following direction-finding problem:

$$\min \nabla f(x_k)^\top d + \mu[\sum y_i + \sum (z_i^+ + z_i^-)]$$
$$s.t.\ \ y \geq g_i(x_k) + \nabla g_i(x_k)^\top d$$
$$z_i^+ - z_i^- = h_i(x_k) + \nabla h_i(x_k)^\top d$$
$$y_i, z_i^+, z_i^- \geq 0$$

- By Theorem 4.2.15, the solution to this program is zero iff x_k is a KKT point.

Some Other Approaches

- Successive Quadratic Programming: Solve the KKT conditions directly using a Newton method.
- This leads to a method which amounts to minimizing a second-order approximation of the Lagrangian.
- From this, we can get a quadratic convergence rate.
- Gradient Projection Method: Recall Homework #4.29. We project the direction $-\nabla f(x_k)$ in order to maintain feasibility.
- This method can be modified to work with nonlinear constraints.