Advanced Optimization

Dr. Alfonso Alba Cadena
fac@fc.uaslp.mx

Facultad de Ciencias
UASLP

Course overview

1. Introduction to optimization
2. Fundamentals of constrained optimization
3. Linear Programming
4. Quadratic Programming
5. Penalization and barrier methods
6. Gradient projection methods
7. Genetic algorithms
8. Neural networks

References

- Numerical Optimization
  Jorge Nocedal and Stephen J. Wright
  Springer

- Introduction to Nonlinear and Global Optimization
  Eligius M.T. Hendrix and Boglárka G.-Tóth
  Springer

Session 1

Introduction

Introduction

- Optimization is the problem of maximizing or minimizing a given function of one or more variables, under some given constraints.

- Optimization is an important issue in many areas. For example:

  - In business: maximize profits and minimize costs.

  - In engineering: maximize performance, minimize complexity.

  - In nature: many phenomena tend towards minimization of energy consumption.

Elements of optimization

- **Objective.-** the function to be optimized. It represents a quantitative measurement of the performance of the system under study.

- **Variables or unknowns.-** the entities or quantities which influence the objective function. The goal is to find the unknowns which maximize or minimize the objective.

- **Constraints.-** the restrictions placed on the variables which must hold for the solution to be considered valid.

The problem of identifying the objective, unknowns, and constraints for a given problem is known as *modeling*.

Mathematical formulation

- In general, an optimization problem can be written as follows:

$$\min_{x \in \mathbb{R}^n} f(x),$$

subject to

$$c_i(x) = 0, \text{ for } i \in \mathcal{E}, \; c_i(x) \geq 0, \text{ for } i \in \mathcal{I}$$

where

  - x is the vector of *variables* or *unknowns*.
  - $f(x) : \mathbb{R}^n \to \mathbb{R}$ is the objective function that we want to maximize or minimize.
  - $c_i : \mathbb{R}^n \to \mathbb{R}$ are the constraint functions which define certain equations that x must satisfy. A point $x \in \mathbb{R}^n$ which satisfies all constraints is called *feasible*.
  - $\mathcal{E}$ is the *set of equality constraints*.
  - $\mathcal{I}$ is the *set of inequality constraints*.

Minimization versus Maximization

- Note that maximizing $f(x)$ is equivalent to minimizing $-f(x)$.

- Without loss of generality, we may assume that the objective function must always be minimized.

Continuous versus Discrete

- Note that we are assuming that $x \in \mathbb{R}^n$.

- In some problems, some of the variables only make sense if they take discrete values, such as integers or labels from a finite set.

- Discrete optimization is not covered in this course, however, some discrete optimization problems can often be posed as continuous optimization problems (e.g., by considering a probability distribution over the discrete set).

Global versus Local optimization

- Most algorithms for nonlinear optimization search only for a local optimum; that is, a point at which the objective function is smaller than all other feasible nearby points.

- The *global solution* is a point with the lowest value of the objective function among **all** feasible points.

- Many successful global optimization algorithms work by solving many local optimization problems. These will be studied in the last part of the course.

Stochastic versus Deterministic

- In some cases, the mathematical model of the problem cannot be fully specified because it may depend on parameters or conditions that are unknown at the time of formulation.

- The uncertainty about these parameters can sometimes be described from a probabilistic point of view.

- *Stochastic optimization* methods incorporate this probabilistic information to generate solutions that optimize the expected performance of the model.

- In contrast, *deterministic optimization* problems are those for which the model is completely known.

Convexity

- A set $S \subset \mathbb{R}^n$ is said to be *convex* if the line segment connecting any two points in S lies entirely inside S. In other words, for any $x, y \in S$, we have that

$$\alpha x + (1 - \alpha)y \in S \quad \text{for all } \alpha \in [0, 1].$$

- A function f is *convex* if its domain S is a convex set and if for any $x, y \in S$ we have that

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y), \ \forall \alpha \in [0, 1].$$

- If the objective function and the feasible region of an optimization problem are both convex, then any local solution of the problem is also a global solution.

Session 2

Basic concepts

General problem

- Let us recall the general formulation of our optimization problem:

$$\min_{x \in \mathbb{R}^n} f(x),$$

subject to

$$c_i(x) = 0, \text{ for } i \in \mathcal{E}, \; c_i(x) \geq 0, \text{ for } i \in \mathcal{I}$$

where

  - x is the vector of *variables* or *unknowns*.
  - $f(x) : \mathbb{R}^n \to \mathbb{R}$ is the objective function that we want to maximize or minimize.
  - $c_i : \mathbb{R}^n \to \mathbb{R}$ are the constraint functions which define certain equations that x must satisfy. A point $x \in \mathbb{R}^n$ which satisfies all constraints is called *feasible*.
  - $\mathcal{E}$ is the *set of equality constraints*.
  - $\mathcal{I}$ is the *set of inequality constraints*.

Feasibility

- The *feasible set* Ω is defined as the set of all points x that satisfy the constraints:

$$\Omega = \{x \mid c_i(x) = 0,\ i \in \mathcal{E}; c_i(x) \geq 0,\ i \in \mathcal{I}\}\,.$$

- The general problem can thus be rewritten as

$$\min_{x \in \Omega} f(x).$$

- If the feasible set is empty, the problem is said to be *infeasible*.

Solutions

- The notion of local and global solutions must now be restricted to feasible points:
  - A vector $\widehat{x}$ is a *local solution* if $\widehat{x} \in \Omega$ and $f(x) \geq f(\widehat{x})$ for all $x \in \mathcal{N} \cap \Omega$, where $\mathcal{N}$ is a neighborhood of $\widehat{x}$.
  - A vector $\widehat{x}$ is a *strict local solution* if $\widehat{x} \in \Omega$ and $f(x) > f(\widehat{x})$ for all $x \in \mathcal{N} \cap \Omega$.
  - If a solution $\widehat{x}$ is the only solution in $\mathcal{N} \cap \Omega$, then $\widehat{x}$ is an *isolated local solution*.
  - A vector $\widehat{x}$ is a *global solution* if $\widehat{x} \in \Omega$ and $f(x) \geq f(\widehat{x})$ for all $x \in \Omega$.

Active constraints

- A constraint $i \in \mathcal{E} \cup \mathcal{I}$ is said to be *active* at a point $x \in \Omega$ if $c_i(x) = 0$.

- Note that equality constraints ($i \in \mathcal{E}$) are always active.

- The *active set* $\mathcal{A}(x)$ at any feasible x is thus defined as

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

Example with one equality constraint

- Consider the problem

$$\min x_1 + x_2, \quad \text{s.t.} \quad x_1^2 + x_2^2 - 2 = 0.$$

  - Write the objective function f and constraint c_1.
  - Plot the constraint c_1 and the level curves of f. Use this plot to find the optimum $\widehat{x}$.
  - Find the normals (gradients) of f and c_1 at $\widehat{x}$ and verify that they are parallel and opposite. Note this does not happen at any other feasible point.

- Consider the *Lagrangian function* for this problem, given by

$$\mathcal{L}(x, \lambda_1) = f(x) - \lambda_1 c_1(x).$$

  A necessary (but not sufficient) optimality condition for this problem can thus be written as

$$\nabla_x \mathcal{L}(\widehat{x}, \widehat{\lambda}_1) = 0.$$

Example with one equality constraint

- Suppose we have a feasible starting point x and want to find a step vector s such that the next iterate, $x + s$ decreases f and remains feasible.

- To retain feasibility we require that $c_1(x + s) = 0$. Applying a first-order Taylor approximation we have

$$c_1(x + s) \approx c_1(x) + \nabla c_1(x)^T s = \nabla c_1(x)^T s = 0,$$

  which means s must be orthogonal to the normal of c_1 at x.

- To produce a decrease in f, we require that

$$0 > f(x + s) - f(x) \approx \nabla f(x)^T s,$$

  or, more compactly, $\nabla f(x)^T s < 0$, which means s must not be on the opposite open half-plane whose normal is $\nabla f(x)$.

- Note that the only way s does not exist (e.g., when x is an optimum) is if $\nabla f(x)$ and $\nabla c_1(x)$ are parallel.

Example with one inequality constraint

- Consider the problem

$$\min x_1 + x_2, \quad \text{s.t.} \quad x_1^2 + x_2^2 - 2 \leq 0.$$

  Note that now $c_1 = 2 - x_1^2 - x_2^2$.

- Consider a feasible x. Under which conditions does a step vector s exists such that $x+s$ decreases f and remains feasible? Consider two cases: (1) when x lies strictly inside the circle (c_1 is inactive) and (2) when x lies in the boundary of the circle (c_1 is active).

- Consider the Lagrangian function previously defined. Note that the optimality conditions reduce to

$$\nabla \mathcal{L}_x(\widehat{x}, \widehat{\lambda}_1) = 0, \ \lambda_1 \geq 0$$

  and

$$\widehat{\lambda}_1 c_1(\widehat{x}) = 0.$$

Example with two constraints

- Consider the problem

$$\min x_1 + x_2, \quad \text{s.t.} \quad 2 - x_1^2 - x_2^2 \leq 0, \; x_2 \geq 0.$$

- Define $c_1(x) = 2 - x_1^2 - x_2^2$, $c_2(x) = x_2$. This time, the Lagrangian is given by

$$\mathcal{L}(x, \lambda) = f(x) - \lambda_1 c_1(x) - \lambda_2 c_2(x),$$

  where $\lambda = (\lambda_1, \lambda_2)^T$ is the vector of Lagrange multipliers.

- The extension of the optimality conditions to this example is

  There exists $\hat{\lambda} \in \mathbb{R}^2$ such that $\nabla_x \mathcal{L}(\hat{x}, \hat{\lambda}) = 0$, $\hat{\lambda} \geq 0$, $\hat{\lambda}^T c(\hat{x}) = 0$.

- Examine these conditions for the following points

  1. $x = (-\sqrt{2}, 0)^T$; that is, the optimum point.
  2. $x = (\sqrt{2}, 0)^T$
  3. $x = (1, 0)^T$

Session 3

Optimality conditions

Optimality conditions

- In the examples shown in the previous session, we established certain conditions that must be met in order to find a step vector s that can improve the current solution x.

- These conditions were obtained from a first-order (linear) Taylor approximation of the objective function and the constraints at the point $x + s$.

- The logic complement of these conditions represent a set of necessary (but not sufficient) conditions that a local optimum $\widehat{x}$ must satisfy. These are called *optimality conditions*.

Feasible directions

- Given a feasible point x, a step vector $d \in \mathbb{R}^n$ is called a *feasible direction* if there exist a sequence of feasible points $z_k \to x$ and a sequence of positive scalars $t_k \to 0$ such that

$$\lim_{k \to \infty} \frac{z_k - x}{t_k} = d.$$

- Let $\mathcal{F}(x)$ be the set of step vectors d such that a new iterate $x + d$ remains feasible according to the first-order approximation of the constraints; that is

$$\mathcal{F}(x) = \left\{ d \;\middle|\; \begin{array}{ll} d^T \nabla c_i(x) = 0, & i \in \mathcal{E}, \\ \\ d^T \nabla c_i(x) \geq 0, & i \in \mathcal{A}(x) \cap \mathcal{I}. \end{array} \right\}.$$

- $\mathcal{F}(x)$ is called the *set of linearized feasible directions* at x.

Constraint qualifications

- In order to generalize the optimality conditions to a larger class of constrained optimization problems, it is required that the set of linearized feasible directions resembles the set of true feasible directions.

- This can be ensured by establishing some qualifications for the constraint functions.

- The most used qualification is the following: Given a point x and the active set $\mathcal{A}(x)$, we say that the *linear independence constraint qualification* (LICQ) holds if the set of active constraint gradients

$$\{\nabla c_i(x) \mid i \in \mathcal{A}(x)\}$$

  is linearly independent.

Lagrangian

- The general form of the Lagrangian function is

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(x),$$

  where λ_i is called the *lagrange multipliers* corresponding to constraint c_i.

- A shorter (vectorial) form of the Lagrangian is

$$\mathcal{L}(x, \lambda) = f(x) - \lambda^T c(x).$$

First order necessary conditions

- Suppose that $\widehat{x}$ is a local solution of the constrained minimization problem, and that the functions f and c_i are continuously differentiable, and that the LICQ holds at $\widehat{x}$. Then there is a Lagrange multiplier vector $\widehat{\lambda}$ such that the following conditions are satisfied:
  - $\nabla_x \mathcal{L}(\widehat{x}, \widehat{\lambda}) = 0$,
  - $c_i(\widehat{x}) = 0$ for all $i \in \mathcal{E}$,
  - $c_i(\widehat{x}) \geq 0$ for all $i \in \mathcal{I}$,
  - $\widehat{\lambda}_i \geq 0$ for all $i \in \mathcal{I}$,
  - $\widehat{\lambda}_i c_i(\widehat{x}) = 0$ for all $i \in \mathcal{E} \cup \mathcal{I}$.
- These are also known as the *Karush-Kuhn-Tucker* (KKT) conditions.

Second order necessary conditions

- Given an optimum point $\widehat{x}$ and the Lagrange multiplier vector $\widehat{\lambda}$ that satisfies the KKT conditions, we define the *critical cone* $\mathcal{C}(\widehat{x}, \widehat{\lambda})$ as

  $$\mathcal{C}(\widehat{x}, \widehat{\lambda}) = \left\{ d \in \mathcal{F}(\widehat{x}) \mid d^T \nabla c_i(\widehat{x}) = 0 \text{ for all } i \in \mathcal{A}(\widehat{x}) \cap \mathcal{I} \text{ with } \widehat{\lambda}_i > 0 \right\}.$$

- In other words, the critical cone is the set of linearized feasible directions that would tend to "adhere" to the active inequality constraints.

- Suppose that $\widehat{x}$ is a local solution satisfying the LICQ condition and $\widehat{\lambda}$ is the Lagrange multiplier vector which satisfies the KKT conditions. Then

  $$d^T \nabla^2_{xx} \mathcal{L}(\widehat{x}, \widehat{\lambda}) d \geq 0,$$

  for all $d \in \mathcal{C}(\widehat{x}, \widehat{\lambda})$.

- In other words, at any local optimum, the Hessian of the Lagrangian has non-negative curvature along critical directions.

Second order sufficient conditions

- Suppose that for some feasible point $\widehat{x} \in \mathbb{R}^n$ there is a Lagrange multiplier vector $\widehat{\lambda}$ such that the KKT conditions are satisfied.

- Suppose also that

$$d^T \nabla^2_{xx} \mathcal{L}(\widehat{x}, \widehat{\lambda}) d > 0,$$

  for all $d \in \mathcal{C}(\widehat{x}, \widehat{\lambda})$, $d \neq 0$.

- Then $\widehat{x}$ is a strict local solution.

Significance of Lagrange multipliers

- The value Lagrange multiplier $\widehat{\lambda}_i$ corresponding to constraint c_i indicates how sensitive the optimal objective value $f(\widehat{x})$ is to the presence of constraint c_i.

- If a constraint c_i is inactive at a local optimum $\widehat{x}$, the corresponding Lagrange multiplier must be zero. This means that $\widehat{x}$ would still be a local minimum even if the constraint was removed.

- Let $\widehat{x}$ be a local solution. We say that an inequality constraint c_i is *strongly active* or *binding* if $i \in \mathcal{A}(\widehat{x})$ and $\widehat{\lambda}_i > 0$ for some $\widehat{\lambda}$ satisfying the KKT conditions.

- An inequality constraint c_i is *weakly active* if $i \in \mathcal{A}(\widehat{x})$ and $\widehat{\lambda}_i = 0$ for all $\widehat{\lambda}$ satisfying the KKT conditions.

Session 4

Elimination of variables

Elimination of variables

- In some cases, it is possible to use the constraints to eliminate some of the variables from the problem, in order to obtain a simpler problem.

- These techniques, however, must be used with care, as they may alter the problem or introduce ill conditioning.

Example: well-applied elimination

- Consider the following problem

$$\min f(x) = f(x_1, x_2, x_3, x_4)$$

subject to

$$x_1 + x_3^2 - x_3 x_4 = 0, \quad \text{and} \quad -x_2 + x_3^2 + x_4 = 0.$$

- Since there is no interaction between x_1 and x_2 in the constraint functions, we can set

$$x_1 = x_3 x_4 - x_3^2, \quad \text{and} \quad x_2 = x_3^2 + x_4,$$

to obtain a new objective function of two variables:

$$h(x_3, x_4) = f(x_3 x_4 - x_3^2, x_3^2 + x_4, x_3, x_4),$$

which can be minimized using unconstrained optimization techniques.

Example: dangerously applied

- Consider now the following example:

$$\min x^2 + y^2 \quad \text{subject to} \quad (x-1)^3 = y^2.$$

- By plotting the constraint function, it can be seen that the solution is $(x, y) = (1, 0)$.

- One may be tempted to eliminate y to obtain the new objective

$$h(x) = x^2 + (x-1)^3,$$

  however $h(x) \to -\infty$ as $x \to -\infty$, thus the new problem is unbounded.

- The problem derives from the fact that the constraint $(x-1)^3 = y^2$ implicitly requires that $x \geq 1$. In fact, the constraint $x \geq 1$ is active at the solution.

- Therefore, if one wishes to eliminate y, then one must explicitly introduce the constraint $x \geq 1$ into the problem.

Linear equality constraints

- Consider the minimization of a nonlinear function subject to a set of linear equality constraints,

$$\min f(x) \quad \text{subject to} \quad Ax = b,$$

where A is an $m \times n$ matrix with $m \leq n$ with full row rank.

- We can find a subset of m linearly independent columns of A. Let P be an $n \times n$ permutation matrix which swaps these columns to the first m column positions in A. Then one can write

$$AP = [B|N],$$

where B is an $m \times m$ matrix formed by these columns and N contains the remaining $n - m$ columns of A.

- We can also define subvectors $x_B \in \mathbb{R}^m$ and $x_N \in \mathbb{R}^{n-m}$ so that

$$P^T x = \begin{bmatrix} x_B \\ x_N \end{bmatrix}.$$

- x_B are called the *basic variables* and B the *basis matrix*.

Linear equality constraints

- Since $PP^T = I$, the constraint $Ax = b$ can be rewritten as

$$b = Ax = AP(P^t x) = Bx_B + Nx_N,$$

  which can be rearranged into

$$x_B = B^{-1}b - B^{-1}Nx_N.$$

- Therefore, the original problem is equivalent to the following unconstrained problem

$$\min_{x_N} h(x_N) = f\left(P\begin{bmatrix} B^{-1}b - B^{-1}Nx_N \\ x_N \end{bmatrix}\right).$$

- This shows that a nonlinear optimization problem with linear equality constraints is equivalent to a unconstrained problem.

Example

- Consider the problem

$$\min \sin(x_1 + x_2) + x_3^2 + \frac{1}{3}\left(x_4 + x_5^4 + \frac{1}{2}x_6\right)$$

subject to

$$\begin{aligned} 8x_1 - 6x_2 + x_3 + 9x_4 + 4x_5 &= 6 \\ 3x_1 + 2x_2 - x_4 + 6x_5 + 4x_6 &= -4. \end{aligned}$$

- Perform elimination of variables as described in the previous slides. Choose the basic variables so that the basis matrix can be easily inverted.

Session 5

Linear Programming

Introduction

- Linear programming (LP) is the problem of finding the optimum of a linear objective function subject to linear equality and inequality constraints.

- LP is the most widely used method of constrained optimization. It has vast applications in many areas such as management, economics, finance, and engineering.

Linear programs

- A linear program is can be usually stated in the following standard form:

$$\min c^T x, \quad \text{subject to } Ax = b,\ x \geq 0,$$

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

Reformulation in the standard form

- Inequality constraints of the form $Ax \leq b$ can be converted to equalities by introducing a vector of *slack variables* z as follows:

$$Ax + z = b,\ z \geq 0.$$

- If some of the variables are allowed to be negative, one can split x into two parts, $x = x^+ - x^-$, where $x^+ = \max(x, 0) \geq 0$ and $x^- = \max(-x, 0) \geq 0$. The problem can thus be written as

$$\min \begin{bmatrix} c \\ -c \\ 0 \end{bmatrix}^T \begin{bmatrix} x^+ \\ x^- \\ z \end{bmatrix}, \quad \text{s.t.} \quad [A \ -A \ I] \begin{bmatrix} x^+ \\ x^- \\ z \end{bmatrix} = b, \quad \begin{bmatrix} x^+ \\ x^- \\ z \end{bmatrix} \geq 0.$$

- Inequality constraints of the form $x \leq u$ or $Ax \geq b$ can also be converted to equalities by adding slack variables:

$$x \leq u \ \leftrightarrow \ x + w = u,\ w \geq 0,$$

$$Ax \geq b \ \leftrightarrow \ Ax - y = b,\ y \geq 0.$$

Solutions

- The objective function is clearly convex since it is linear. Its contours are hyperplanes. The feasible region is a convex polytope.

- A linear program can have
  - No solution if the feasible region is empty (the *infeasible* case).
  - No solution if the objective function is unbounded below on the feasible region (the *unbounded* case).
  - A unique solution located at a vertex of the feasible polytope.
  - Infinite solutions, where the set of optimal points is an edge, a face, or the entire feasible set.

Optimality conditions

- Only the first-order conditions (the KKT conditions) are required for a point x to be optimal. Due to convexity, these conditions ensure that the optimum is global.

- It can be proven that if the constraints $c_i(x)$ are all linear, then the set of linearized feasible directions $\mathcal{F}(x)$ is equal to the set of feasible directions. Therefore, the LICQ is not required for linear programs.

- Let $[\lambda, s]^T$ be the vector of Lagrange multipliers, where $\lambda \in \mathbb{R}^m$ are the Lagrange multipliers which correspond to the equality constraints $Ax = b$, while $s \in \mathbb{R}^n$ corresponds to the bound constraints $x \geq 0$. The Lagrangian function for this problem is therefore given by

$$\mathcal{L}(x, \lambda, s) = c^T x - \lambda^T (Ax - b) - s^t x.$$

Optimality conditions

- The first order (Karush-Kuhn-Tucker) conditions for $\widehat{x}$ to be a solution of the linear program are that there exist vectors $\widehat{\lambda}$ and $\widehat{s}$ such that
  - $A^T\widehat{\lambda} + \widehat{s} = c$,
  - $A\widehat{x} = b$,
  - $\widehat{x} \geq 0$,
  - $\widehat{s} \geq 0$,
  - $\widehat{x}_i\widehat{s}_i = 0,\ i = 1, 2, \ldots, n.$
- These conditions can also be rewritten as

$$c^T\widehat{x} = (A^T\widehat{\lambda} + \widehat{s})^T\widehat{x} = (A\widehat{x})^T\widehat{\lambda} = b^T\widehat{\lambda}.$$

Geometry of the feasible set

- Without loss of generality, it can be assumed that the matrix A has full row rank.

- A vector x is a *basic feasible point* if it is feasible and if there exists a subset $\mathcal{B}$ of the index set $\{1, 2, \ldots, n\}$ such that
  - $\mathcal{B}$ contains exactly m indices.
  - $i \notin \mathcal{B} \;\Rightarrow\; x_i = 0$; that is, the bound $x_i \geq 0$ can be inactive only if $i \in \mathcal{B}$.
  - The $m \times m$ matrix B defined by

    $$B = [A_i]_{i \in \mathcal{B}}$$

    is nonsingular, where A_i represents the i-th column of A.

  A set $\mathcal{B}$ satisfying these properties is called a *basis* and the corresponding matrix B is called the *basis matrix*.

Geometry of the feasible set

- If the feasible region is nonempty, then there is at least one basic feasible point.

- If the problem has solutions, then at least one solution is a basic feasible point.

- The basic feasible points are vertices of the feasible polytope $\{x \mid Ax = b,\ x \geq 0\}$.

- A basis $\mathcal{B}$ is said to be degenerate if $x_i = 0$ for some $i \in \mathcal{B}$, where x is the basic feasible point corresponding to $\mathcal{B}$. A linear program is said to be degenerate if it has at least one degenerate basis.

Session 6

The Simplex Method

Introduction

- The simplex method is an iterative algorithm to solve linear programs.

- All iterates of this method are basic feasible points, and therefore vertices of the feasible polytope.

- The algorithm starts with some vertex as initial solution and on most steps moves from one vertex to an adjacent one for which the basis $\mathcal{B}$ differs in exactly one component.

- On most steps, the value of the objective function $c^T x$ is decreased.

Variable separation

- From the KKT conditions and the basis $\mathcal{B}$ one can obtain the values of the variables x, and the dual variables (λ, s).

- Let $\mathcal{N} = \{1, \ldots, n\} \setminus \mathcal{B}$ be the *nonbasic index set*. The nonbasic matrix N is given by $N = [A_i]_{i \in \mathcal{N}}$.

- The n-element vectors x, s and c can be partitioned according to the sets $\mathcal{B}$ and $\mathcal{N}$ as follows:

$$\begin{array}{ll} x_B = [x_i]_{i \in \mathcal{B}}, & x_N = [x_i]_{i \in \mathcal{N}} \\ s_B = [s_i]_{i \in \mathcal{B}}, & s_N = [s_i]_{i \in \mathcal{N}} \\ c_B = [c_i]_{i \in \mathcal{B}}, & c_N = [c_i]_{i \in \mathcal{N}} \end{array}$$

Variable separation

- The second KKT condition states that

$$Ax = Bx_B + Nx_N = b.$$

- Suppose x is a basic feasible point. By definition, $x_N = 0$, therefore $x_B = B^{-1}b$. Clearly, the nonnegativity condition $x \geq 0$ is also satisfied.

- s is chosen to satisfy the complementarity condition $x_i s_i = 0$ by setting $s_B = 0$.

- The first KKT condition can be partitioned into

$$B^T\lambda = c_B, \quad N^T\lambda + s_N = c_N,$$

  from which $\lambda = B^{-T}c_B$ and

$$s_N = c_N - N^T\lambda = c_N - (B^{-1}N)^T c_B.$$

Change of basis

- The only KKT condition which has not been explicitly enforced is $s \geq 0$. Our choice for x_B satisfies this condition. If the vector $s_N \geq 0$, then an optimal solution has been found.

- If one or more of the components of s_N is negative, we chose one of their corresponding indexes q in $\mathcal{N}$ (for which $s_q < 0$) to enter the basis $\mathcal{B}$. This is called the *entering index*.

- It can be shown that the objective $c^T x$ will decrease if and only if

  1. $s_q < 0$, and
  2. x_q can be increased away from zero while maintaining feasibility.

- Since the size of $\mathcal{B}$ must remain constant, including q in $\mathcal{B}$ requires one of the indices $p \in \mathcal{B}$ to leave the basis.

Pivoting

- The process of selecting the entering index q and the leaving index p is known as *pivoting*. This process is described as follows:

  1. Choose $q \in \mathbb{N}$ such that $s_q < 0$, and allow x_q to increase from zero.

  2. Fix all other components of x_N at zero, and figure the effect of increasing x_q on the current basis vector x_B, considering that we want to stay in the feasible region determined by the equality constraints $Ax = b$.

  3. Keep increasing x_q until one of the components of x_B (say, x_p) becomes zero, or until one determines that no such component exists (the unbounded case).

  4. Remove index p (the leaving index) from $\mathbb{B}$ and replace it with the entering index q.

Pivoting

- Let x be the current solution and $\tilde{x}$ be the new iterate. Since $Ax = A\tilde{x} = b$ and $x_N = 0$ and $\tilde{x}_i = 0$ for $i \in \mathcal{N} \setminus \{q\}$, then

$$A\tilde{x} = B\tilde{x}_B + A_q\tilde{x}_q = Bx_B = Ax.$$

  Multiplying by B^{-1} and rearranging we obtain

$$\tilde{x}_B = x_B - B^{-1}A_q\tilde{x}_q.$$

- Increasing x_q eventually leads a new constraint $x_p \geq 0$ to become active, unless the problem is unbounded. In this case $\tilde{x}_B = x_B - B^{-1}A_q\tilde{x}_q \geq 0$ holds for all positive values of $\tilde{x}_q$. This happens when $B^{-1}A_q \leq 0$.

The Simplex method

- Given $\mathcal{B}$, $\mathcal{N}$, $x_B = B^{-1}b \geq 0$, and $x_N = 0$:
- Solve $B^T\lambda = c_B$ for λ.
- Compute $s_N = c_N - N^T\lambda$.
- If $s_N \geq 0$, terminate the algorithm and return x as the optimal point.
- Select $q \in \mathcal{N}$ with $s_q < 0$ (the entering index).
- Solve $Bd = A_q$ for d.
- If $d \leq 0$, terminate the algorithm since the problem is unbounded.
- Calculate $\tilde{x}_q = \min_{i|d_i>0}(x_B)_i/d_i$ and let p denote the minimizing i.
- Update $\tilde{x}_B = x_B - d\tilde{x}_q$, $\tilde{x}_N = (0, \ldots, 0, \tilde{x}_q, 0, \ldots, 0)^T$.
- Change $\mathcal{B}$ by adding q and removing the basic variable corresponding to column p of B.

Example

- Solve the problem

$$\min \ -4x_1 - 2x_2$$

subject to

$$\begin{aligned} x_1 + x_2 + x_3 &= 5, \\ 2x_1 + \tfrac{1}{2}x_2 + x_4 &= 8, \\ x &\geq 0. \end{aligned}$$

using $\mathcal{B} = \{3, 4\}$ as initial basis. Note that the solution is $x = (11/3, 4/3, 0, 0)$.

Session 7

Simplex implementation details

Selection of the entering index

- Usually, there are many negative components of s_N at each step. The selection of the entering index among these components may have a significant impact in the convergence speed of the simplex method.

- Many strategies for selecting the entering index have been devised. However, the computational cost of finding a good entering index might be sometimes higher than simply taking a longer path towards the optimum.

Selection strategies

- **Dantzig's rule.-** Choose q such that s_q is the most negative component of $s_N = N^T\lambda$.

- **Partial pricing.-** Calculate only a subvector of s_N and select q among the negative components of the subvector. To give all indices in $\mathcal{N}$ a chance to enter the basis, this strategy must cycle through all the non-basic elements.

- **Multiple pricing.-** Evaluate s_q for all q in a small subset $\mathcal{S} \subset \mathcal{N}$ and for each $s_q < 0$ find the maximum value of $\tilde{x}_q$ which maintains feasibility and the corresponding change in the objective given by $s_q\tilde{x}_q$. The process is repeated until s_q are nonnegative for all $q \in \mathcal{S}$. Then s_N is computed and a new subset $\mathcal{S}$ is chosen.

Finding an initial basic solution

- Finding an initial basis is usually a non-trivial problem and its difficulty is equivalent to solving a linear program.

- Most implementations use a two-phase approach where the first phase consists in solving a linear program specifically designed to find an initial solution for the original problem, which is then solved in phase 2.

- The Phase I problem is as follows:

$$\min e^T z, \quad \text{subject to } Ax + Ez = b,\ x \geq 0,\ z \geq 0,$$

  where $z \in \mathbb{R}^m$, $e = (1, 1, \ldots, 1)^T$, and E is a diagonal matrix whose elements are $E_jj = 1$ if $b_j \geq 0$ and $E_jj = -1$ if $b_j < 0$. It is easy to see that the point given by $x = 0$, $z_j = |b_j|$ is a basic feasible solution for this problem (the initial basis matrix is $B = E$).

- The Phase I problem has an optimal solution $(\hat{x}, \hat{z})$ with $\hat{z} = 0$ if and only if the original problem has at least one feasible point.

Finding an initial basic solution

- The Phase II problem is given by

$$\min c^T x, \quad \text{subject to } Ax + z = b,\ x \geq 0,\ 0 \geq z \geq 0.$$

- Note that (1) this problem is equivalent to the original problem, since z is constrained to be zero at all times, and (2) the solution $(\widehat{x}, \widehat{z})$ of the Phase I problem is a basic feasible point for the Phase II problem (with the same basis as in the Phase I problem).

- The simplex implementation for Phase II can delete those components of z which leave the basis (and their corresponding columns in the coefficient matrix) in order to increase the efficiency and robustness of the procedure.

Session 8

Quadratic Programming

Quadratic programs

- A *quadratic program* (QP) is an optimization problem with a quadratic objective function and linear constraints. The general form of a QP is

$$\begin{array}{rrcl} \min_x & q(x) & = & \frac{1}{2}x^T G x + x^T c \\ \text{subject to} & a_i^T x & = & b_i,\ i \in \mathcal{E}, \\ & a_i^T x & \geq & b_i,\ i \in \mathcal{I}. \end{array}$$

- If the Hessian matrix G is positive semidefinite, the QP is said to be convex and it is similar in difficulty to a linear program. On the other hand, nonconvex QPs, in which G is an indefinite matrix, can have several stationary points and local minima.

Equality-constrained quadratic programs

- A special case of QPs is obtained when there are no inequality constraints. In this case, the problem can be stated as follows:

$$\min q(x) = \frac{1}{2} x^T G x + x^T c, \text{ subject to } Ax = b,$$

  where A is an $m \times n$ Jacobian matrix of the constraints. This matrix can be assumed to have full row rank (rank m).

- Although this sub-class of QPs seems very limited, we will later see that some algorithms for general QPs require to solve an equality-constrained QP at each iteration.

Null space of the constraint Jacobian

- Without loss of generality, one can assume that the first m columns of A are linearly independent, so that the matrix A and the variable vector x can be expressed as

$$A = [B \mid N], \quad x = \begin{bmatrix} x_B \\ x_N \end{bmatrix},$$

where B is a basis matrix, N is the non-basic matrix, x_B are the basic variables, and x_N the non-basic variables.

- Recall also that x_B can be expressed in terms of x_N:

$$x_B = B^{-1}b - B^{-1}Nx_N.$$

- Therefore, the variable vector x can be written as

$$x = Yb + Zx_N, \; Y = \begin{bmatrix} B^{-1} \\ 0 \end{bmatrix}, \; Z = \begin{bmatrix} -B^{-1}N \\ I \end{bmatrix}.$$

- The matrix Z has $n - m$ linearly independent columns and satisfies $AZ = 0$, therefore, Z is a basis for the null space of A.

First-order necessary conditions

- The KKT conditions for $\hat{x}$ to be a solution of the equality-constrained QP state that there exists a vector $\hat{\lambda}$ of Lagrange multipliers such that the following equation is satisfied:

$$\begin{bmatrix} G & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} -c \\ b \end{bmatrix}.$$

- Expressing the solution as $\hat{x} = x + p$ where x is some estimate of the solution and p is the required step vector, the system above can be rewritten as

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -p \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} Gx + c \\ Ax - b \end{bmatrix}.$$

- The matrix $K = \begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix}$ is called the Karush-Kuhn-Tucker (KKT) matrix.

Nonsingularity of the KKT matrix

- If A has full row rank and the reduced-Hessian matrix $Z^T G Z$ is positive definite, then the KKT matrix

$$K = \begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix}$$

  is nonsingular, and hence the equality-constrained QP has a unique solution $(\widehat{x}, \widehat{\lambda})$.

- Moreover, if the above conditions are satisfied, then $\widehat{x}$ is a unique global solution of the equality-constrained QP.

- When the reduced Hessian $Z^T G Z$ is positive semidefinite with zero eigenvalues, the vector $\widehat{x}$ satisfying the KKT system is a local minimizer but not a strict local minimizer. If the reduced Hessian has negative eigenvalues, when $\widehat{x}$ is only a stationary point, but not a local minimizer.

Session 9

Solving the KKT system

Indefiniteness of the KKT matrix

- Let us recall that the solution of the equality-constrained quadratic program

$$\min q(x) = x^T G x + x^T c, \quad \text{subject to } Ax = b,$$

where A is an $m \times n$ matrix with rank m, can be obtained as $\widehat{x} = x + p$, where x is an initial solution, and p is a step vector which must satisfy the KKT system

$$K \begin{bmatrix} -p \\ \widehat{\lambda} \end{bmatrix} = \begin{bmatrix} Gx + c \\ Ax - b \end{bmatrix},$$

where the KKT matrix K is given by

$$K = \begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix}.$$

- It can be proven that if $Z^T G Z$ is positive definite, then K is indefinite for all $m \geq 1$, therefore, the KKT system cannot be solved using Cholesky factorization.

Solutions for symmetric Hessian

- If the Hessian matrix G is symmetric, then the KKT matrix K is also symmetric. In this case, K has a factorization of the form

$$P^T K P = L T L^T,$$

where P is a permutation matrix, L is a lower triangular matrix, and T is a symmetric tri-diagonal matrix made from either 1×1 or 2×2 blocks.

- Once the factorization is obtained, the solution $[p, \widehat{lambda}]^T$ can be obtained from the following sequence of operations:

  1. Solve $Lz_1 = P^T \begin{bmatrix} Gx + c \\ Ax - b \end{bmatrix}$ to obtain z_1.
  2. Solve $Tz_2 = z_1$ to obtain z_2.
  3. Solve $L^T z_3 = z_2$ to obtain z_3.
  4. Let $[-p, \widehat{\lambda}]^T = Pz_3$.

  The solution of $Tz_2 = z_1$ requires solving a number of smaller 1×1 and 2×2 systems.

Schur-Complement Method

- Consider the case where the Hessian matrix G is symmetric and positive definite.

- Multiplying the first equation of the KKT system by AG^1 and substituting the second equation one can obtain a linear system for $\hat{\lambda}$ alone:

$$(AG^{-1}A^T)\hat{\lambda} = (AG^{-1}g - h).$$

  Since G is symmetric and positive definite, then $AG^{-1}A^T$ is also symmetric positive definite, and therefore the system can be solved efficiently for $\hat{\lambda}$ using Cholesky decomposition.

- Once $\hat{\lambda}$ is known, p can be recovered from the first equation by solving

$$Gp = A^T\hat{\lambda} - g$$

  also using Cholesky decomposition.

Null-space method

- Suppose the matrices Y and Z (slide 63) are known. Then, p can be partitioned into two components p_Y and p_Z such that $p = Yp_Y + Zp_Z$.

- Substituting p into the second equation of the KKT system, and recalling that $AZ = 0$, we obtain $(AY)p_Y = b - Ax$.

- Since A has rank m and $[Y|Z]$ is a nonsingular $n \times n$ matrix, then the product $A[Y|Z] = [AY|O]$ has rank m. Therefore AY must be a nonsingular $m \times m$ matrix and p_Y can be uniquely determined.

- Substituting p into the first equation of the KKT system yields

$$-GYp_Y - GZp_Z + A^T\hat{\lambda} = Gx + c.$$

  Multiplying by Z^T and rearranging one obtains

$$(Z^TGZ)p_Z = -Z^TGYp_Y - Z^T(Gx + c),$$

  which can be efficiently solved for p_Z by performing a Cholesky decomposition of the reduced-Hessian Z^TGZ.

Session 10

Inequality-constrained quadratic programs

Inequality-constrained quadratic programs

- Consider the general quadratic program:

$$\begin{array}{rrcl} \min_x & q(x) & = & \frac{1}{2}x^T Gx + x^T c \\ \text{subject to} & a_i^T x & = & b_i,\ i \in \mathcal{E}, \\ & a_i^T x & \geq & b_i,\ i \in \mathcal{I}. \end{array}$$

- The Lagrangian function for this problem is

$$\mathcal{L}(x, \lambda) = \frac{1}{2}x^T Gx + x^T c - \sum_{i \in \mathcal{I} \cup \mathcal{E}} \lambda_i (a_i T x - b_i).$$

- The active set $\mathcal{A}(x)$ is given by

$$\mathcal{A}(x) = \{i \in \mathcal{E} \cup \mathcal{I} \mid a_i^T x = b_i\}.$$

- We will limit our study to convex QPs; that is, QPs for which G is positive definite.

Active-set methods for convex QPs

- Suppose the optimal solution $\widehat{x}$ is unknown, but the optimal active set $\mathcal{A}(\widehat{x})$ is known in advance. In this case, one could find the solution $\widehat{x}$ by solving the following equality-constrained QP:

  $$\min_x q(x) = \frac{1}{2}x^T G x + x^T c, \quad \text{subject to} \quad a_i^T x = b_i, \; i \in \mathcal{A}(\widehat{x}).$$

- Therefore, one of the main challenges in solving general QPs is determining the optimal active set.

- Active set methods start with an initial guess of $\mathcal{A}(\widehat{x})$ and use gradient and Lagrange multiplier information to include and exclude indices from $\mathcal{A}(\widehat{x})$ until optimality is detected.

Active-set for QP vs Simplex

- The simplex method used for solving linear programs is one example of an active-set method.

- The current guess for $\mathcal{A}(x)$ in the simplex method is union of the set of non-basic indices $\mathcal{N}$ and $\mathcal{E}$.

- Active-set methods for QPs differ from the simplex method in that the iterates and the optimal solution are not necessarily vertices of the feasible region. Therefore, the size of the active set is not necessarily constant.

Working set

- At each step of the active-set method, one must solve a quadratic sub-problem in which some of the inequality constraints, along with the equality constraints, are imposed as equalities, and the other inequality constraints are disregarded.

- The set of imposed equality constraints for the sub-problem is known as the *working set*, and is denoted by $\mathcal{W}_k$ for the k-th iterate x_k.

- One important requirement that must be satisfied is that the gradients a_i of the constraints in $\mathcal{W}_k$ should be linearly independant, even if the full set of active constraints at x_k has linearly dependant gradients.

Quadratic sub-problems

- Consider an iterate x_k and the working set $\mathcal{W}_k$, and suppose x_k does not minimize q in the subspace defined by the constraints in the working set.

- To improve the current solution, one can solve an equality constrained QP where the constrains in $\mathcal{W}_k$ are imposed as equalities. The solution x of this problem can be expressed in terms of a step vector p such that $x = x_k + p$, so that

$$q(x) = q(x_k + p) = \frac{1}{2} p^T G p + g_k^t p + \rho_k,$$

  where $g_k = G x_k + c$ and $\rho_k = \frac{1}{2} x_k^T G x_k + c^T x_k$. Since ρ_k does not depend on p, we can remove it from the objective function without affecting the solution of this sub-problem.

- Therefore, the problem to solve at the k-th iteration is

$$\min_p \frac{1}{2} p^T G p + g_k^T p, \quad \text{subject to} \quad a_i^T p = 0, \ i \in \mathcal{W}_k.$$

- The solution of this problem will be denoted by p_k.

Computing the next iterate

- If the optimal p_k obtained from the k-th sub-problem is nonzero, then moving along the direction p_k will improve the objective.

- If $x_k + p_k$ is feasible (with respect to all the constraints), then we set $x_{k+1} = x_k + p_k$; otherwise, we set

  $$x_{k+1} = \alpha_k p_k,$$

  where α_k is the largest value in $[0, 1]$ for which all constraints are satisfied. This value can be computed as

  $$\alpha_k = \min\left(1, \min_{i \notin \mathcal{W}_k,\ a_i^T p_k < 0} \frac{b_i - a_i^T x_k}{a_i^T p_k}\right).$$

- The constraints i for which the minimum in the right-hand side of the previous equation is achieved are called the *blocking constraints*.

- If $\alpha_k < 1$, the step along p_k was blocked by some constraint not in $\mathcal{W}_k$. A new working set $\mathcal{W}_{k+1}$ is constructed by adding one of the blocking constraints to $\mathcal{W}_k$.

Detecting optimality

- The algorithm continues iterating as previously shown until an iterate $\hat{x}$ minimizes q over the current working set; in other words, until the solution of the sub-problem is $p = 0$.

- In this case, $p = 0$ satisfies the second equation of the KKT system, and the first equation amounts to $\sum_{i \in \hat{\mathcal{W}}} a_i \hat{\lambda}_i = G\hat{x} + c$.

- If one sets the Lagrange multipliers corresponding to inequality constraints not in $\hat{\mathcal{W}}$ to be zero, then $\hat{x}$ and $\hat{\lambda}$ satisfy the first KKT condition for the original QP with inequality constraints. Because of the control imposed on the step length, $\hat{x}$ also satisfies the 2nd and 3rd (feasibility) KKT conditions.

- If the Lagrange multipliers $\hat{\lambda}_i$ for $i \in \hat{\mathcal{W}} \cap \mathcal{I}$ are all non-negative, then the fourth KKT condition is also satisfied and $\hat{x}$ is a global solution of the original QP (due to the QP being convex).

- If one or more multipliers $\hat{\lambda}_j$, $j \in \hat{\mathcal{W}} \cap \mathcal{I}$ are negative, the objective function may be further decreased by removing one of these indices from the working set and solving a new sub-problem.

Active-set algorithm

- Compute a feasible starting point x_0 and set $\mathcal{W}_0$ to be a subset of the active constraints at x_0.
- For $k = 0, 1, 2, \ldots$
  - Solve the equality-constrained sub-problem to find p_k.
  - If $p_k = 0$
    - Compute the Lagrangian multipliers that satisfy $\sum_{i \in \mathcal{W}_k} a_i \widehat{\lambda}_i = Gx_k + c$.
    - If $\widehat{\lambda}_i \geq 0$ for all $i \in \mathcal{W}_k \cap \mathcal{I}$, stop with x_k as global solution.
    - Else, let $j = \arg\min_{i \in \mathcal{W}_k} \widehat{\lambda}_i$, $x_{k+1} = x_k$, and $\mathcal{W}_{k+1} = \mathcal{W}_k \setminus \{j\}$.
  - Else
    - Compute α_k as given in slide 77, and let $x_{k+1} = x_k + \alpha_k p_k$.
    - If there are blocking constraints ($\alpha_k < 1$), obtain $\mathcal{W}_{k+1}$ by adding one of the blocking constraints to $\mathcal{W}_k$; otherwise, let $\mathcal{W}_{k+1} = \mathcal{W}_k$.

Session 11

Gradient Projection for bound-constrained QPs

Gradient projection method

- In contrast to the active sets method, the gradient projection method allows the active set to change rapidly at each iteration. It is most efficient when the constraints are simple, for example, when there are only bounds on the variables.

- Consider the bound-constrained quadratic problem;

$$\min_x q(x) = \frac{1}{2}x^T G x + x^T c, \quad \text{subject to } l \leq x \leq u,$$

  where G is symmetric and l and u are vectors of lower and upper bounds on the components of x, with $l_i < u_i$ for all i. In this case, we do not require G to be positive definite (i.e., the problem can be non-convex).

- If one of the variables is unbounded, we set the corresponding component of l or u to $-\infty$ or $+\infty$, respectively.

General idea

- Each iteration of the gradient projection method consists of two stages.

- Let x be the current solution. In the first stage, one searches along the direction of steepest descent $-g$, where $g = Gx + c$. When a bound is encountered, the search direction is bent so that it stays feasible. This results in a piecewise linear path along which the minimizer x^c of q is searched. This minimizer is known as the *Cauchy point*.

- In the second stage, the working set is now defined as the set of constraints active at x^c, and then solve a subproblem in which the variables involved in the active constraints are fixed at their corresponding values in x^c.

Projected descent path

- The projection $P(x, l, u)$ of an arbitrary point x onto the feasible region can be defined as follows. The i-th component of the projection is given by

$$P(x, l, u)_i = \begin{cases} l_i & \text{if } x_i < l_i, \\ x_i & \text{if } l_i \leq x_i \leq u_i, \\ u_i & \text{if } x_i > u_i. \end{cases}$$

- The piece-wise linear path obtained by projecting the steepest descent direction at x onto the feasible region is $x(t) = P(x - tg, l, u)$.

Path segments

- To find the Cauchy point, one must find the first local minimizer of q along this path; that is, the first local minimizer of the univariate function $q(x(t))$. Since this function is only piecewise differentiable, one must find first the singular points; that is, those points along the steepest descent direction where each bound becomes active. These are given by

$$\bar{t}_i = \begin{cases} (x_i - u_i)/g_i & \text{if } g_i < 0 \text{ and } u_i < +\infty, \\ (x_i - l_i)/g_i & \text{if } g_i > 0 \text{ and } l_i > -\infty, \\ \infty & \text{otherwise.} \end{cases}$$

- To search for the first local minimizer along $x(t)$, we eliminate the duplicate values and zero values of $\bar{t}_i$ and order the remaining values increasingly to obtain a reduced, sorted set $\{t_1, t_2, \ldots, t_m\}$ with $t_0 = 0 < t_1 < t_2 < \ldots < t_m$. Each segment $[t_{k-1}, t_k]$ must be examined in succession to determine if it contains a local minimizer.

Minimization of each segment

- Suppose the next segment to examine is $[t_{j-1}, t_j]$. We have that

$$x(t) = x(t_j - 1) + (\Delta t)p_{j-1},$$

where $\Delta t = t - t_{j-1} \in [0, t_j - t_{j-1}]$ and

$$p_i^{j-1} = \begin{cases} -g_i & \text{if } t_{j-1} < \bar{t}_i, \\ 0 & \text{otherwise.} \end{cases}$$

- Then, the objective function along the segment can be written as

$$q(x(t)) = f_{j-1} + f'_{j-1}\Delta t + \frac{1}{2}f''_{j-1}(\Delta t)^2, \quad \Delta t \in [0, t_j - t_{j-1}],$$

where

$$\begin{aligned} f_{j-1} &= c^T x(t_{j-1}) + \tfrac{1}{2}x(t_{j-1})^T G x(t_{j-1}), \\ f'_{j-1} &= c^T p^{j-1} + x(t_{j-1})^T G p^{j-1}, \\ f''_{j-1} &= (p^{j-1})^T G p^{j-1}. \end{aligned}$$

Solution of the quadratic polynomial

- The solution of the quadratic univariate objective function for interval $[t_{j-1}, t_j]$,

$$q(x(t)) = f_{j-1} + f'_{j-1}\Delta t + \frac{1}{2} f''_{j-1}(\Delta t)^2, \quad \Delta t \in [0, t_j - t_{j-1}],$$

  is given by $\Delta \widehat{t} = -f'_{j-1}/f''_{j-1}$.

- The following cases may occur

  1. If $f'_{j-1} > 0$ there is a local minimizer of $q(x(t))$ at $t = t_{j-1}$
  2. If $\Delta \widehat{t} \in [0, t_j - t_{j-1})$ there is a minimizer at $t = t_{j-1} + \Delta \widehat{t}$.
  3. In any other case, the minimizer of $q(x(t))$ does not belong to the current segment, so we move on to the next interval $[t_j, t_{j+1}]$ and continue the search.

Subspace minimization

- Once the Cauchy point x^c has been found, the active set at this point is defined by

$$\mathcal{A}(x^c) = \{i \mid x_i^c = l_i \text{ or } x_i^c = u_i\}.$$

- The second stage of the gradient projection method consists in approximately solving the QP obtained by fixing the components x_i for $i \in \mathcal{A}(x^c)$ at the values x_i^c. This subproblem can be formulated as

$$\begin{array}{rl} \min_x & q(x) = \frac{1}{2}x^T G x + x^T c, \\ \text{subject to} & x_i = x_i^c, \ i \in \mathcal{A}(x^c), \\ & l_i \leq x_i \leq u_i, \ i \notin \mathcal{A}(x^c). \end{array}$$

- It is not necessary to solve this problem exactly (particularly since this subproblem may be as difficult as the original). It is only required that the solution $\tilde{x}$ of this subproblem is feasible and satisfies $q(\tilde{x}) \leq q(x^c)$.

- For example, one could eliminate the inequality constraints in the subproblem and apply an unconstrained iterative method to solve the resulting problem, and terminate as soon as a bound $l_i \leq x_i \leq u_i$, $i \notin \mathcal{A}(x^c)$ is encountered.

Gradient projection algorithm

- Compute a feasible starting point x^0.
- For $k = 0, 1, 2, \ldots$
  - If x^k satisfies the KKT conditions, stop with $\widehat{x} = x^k$ as solution.
  - Otherwise, set $x = x^k$ and find the Cauchy point x^c.
  - Find an approximate solution $\tilde{x}$ in the subspace defined by the active constraints $\mathcal{A}(x^c)$ at x^c such that $q(\tilde{x}) \leq q(x^c)$ and $\tilde{x}$ is feasible.
  - Set $x^{k+1} = \tilde{x}$.

Session 13

Penalty Methods

Introduction

- Penalty methods attempt to solve a constrained optimization problem by replacing the original problem with a sequence of sub-problems in which the constraints are represented by terms added to the objective. Each of the sub-problems can be solved by common unconstrained techniques.

- The most common penalty terms are:
  - **Quadratic penalty:** The penalty terms are the squares of the violations of each constraint (e.g., $c_i^2(x)$ for $i \in \mathcal{E}$).
  - **Nonsmooth exact penalty:** Often represented as the ℓ_1 norm of the violations.
  - **Augmented Lagrangian:**

Quadratic Penalty for equality-constrained problems

- Consider the equality-constrained problem

$$\min_x f(x), \quad \text{subject to} \quad c_i(x) = 0, \ i \in \mathcal{E}.$$

- The quadratic penalty function $Q(x, \mu)$ for this formulation is defined as

$$Q(x, \mu) = f(x) + \frac{\mu}{2} \sum_{i \in \mathcal{E}} c_i^2(x),$$

  where $\mu > 0$ is the *penalty parameter*.

- By driving $\mu \to \infty$, we penalize the constraint violations with increasing severity and force the minimizer of $Q(x, \mu)$ closer to the feasible region of the original constrained problem.

- The quadratic penalty method considers a sequence of values $\{\mu_k\}$ where $\mu_k \to \infty$ as $k \to \infty$, and searches for an approximate minimizer x_k of $Q(x, \mu_k)$ for each k. Since Q is smooth, one can use most unconstrained optimization techniques.

Quadratic Penalty for inequality constraints

- For the general constrained problem

$$\min_x f(x), \quad \text{subject to} \quad c_i(x) = 0,\ i \in \mathcal{E}, \quad c_i(x) \geq 0,\ i \in \mathcal{I},$$

  the quadratic penalty function can be defined as

$$Q(x, \mu) = f(x) + \frac{\mu}{2} \sum_{i \in \mathcal{E}} c_i^2(x) + \frac{\mu}{2} \sum_{i \in \mathcal{I}} \left([c_i(x)]^- \right)^2,$$

  where $[y]^- = \max(-y, 0)$.

- In this case, $Q(x, \mu)$ may be nonsmooth so more sophisticated unconstrained minimization methods may be required.

Quadratic Penalty Algorithm

- Given $\mu_0 > 0$, a starting point x_0^s, and a nonnegative sequence $\{\tau_k\}$ such that $\tau_k \to 0$,

- For $k = 0, 1, 2, \ldots$
  - Find an approximate minimizer x_k of $Q(x, \mu_k)$ using x_k^s as starting point, and terminating when $||\nabla_x Q(x, \mu_k)|| \leq \tau_k$.
  - If final convergence test is satisfied, then stop with approximate solution x_k.
  - Otherwise, choose a new penalty parameter $\mu_{k+1} > \mu_k$ and a new starting point x_{k+1}^s (e.g., $x_{k+1}^s = x_k$).

- Note that, as μ_k becomes large, the minimization of $Q(x, \mu_k)$ may become more difficult since the Hessian $\nabla_{xx}^2 Q(x, \mu_k)$ becomes ill conditioned (i.e., it becomes arbitrarily large).

Session 14

Nonsmooth exact penalty methods

Motivation

- One disadvantage of sequential penalty methods (such as the quadratic penalty method) is that theoretically they need to solve an infinite number of unconstrained optimization problems to guarantee feasibility.

- In practice, one can project the solution to the feasible region; however, a large number of iterations may still be required.

- In contrast, exact penalty methods avoid this long sequence by using *exact penalty functions*. A penalty function is exact if the solution of the penalty problem is also a solution of the original problem for a finite value of the penalty parameter μ.

- The disadvantage of using exact penalty functions is that they are not differentiable.

Exact penalty functions

- The most common exact penalty function is the ℓ_1 norm of the constraint violations. In this case, the new objective function may be defined as

$$Q_1(x, \mu) = f(x) + \mu \sum_{i \in \mathcal{E}} |c_i(x)| + \mu \sum_{i \in \mathcal{I}} [c_i(x)]^-.$$

- Note that we can define $c_{\mathcal{E}}(x)$ as the vector whose components are $c_i(x)$, $i \in \mathcal{E}$ and $[c_{\mathcal{I}}(x)]^-$ as the vector whose components are $[c_i(x)]^-$, $i \in \mathcal{I}$. Therefore,

$$Q_1(x, \mu) = f(x) + \mu ||c_{\mathcal{E}}(x)||_1 + \mu ||[c_{\mathcal{I}}(x)]^-||_1,$$

  where $|| \cdot ||_n$ denotes the ℓ_n norm.

- In general, any vector norm can be used (with varying results) to measure infeasibility.

Classical ℓ_n penalty method

- Define the infeasibility measure

$$h_n(x_k) = ||c_{\mathcal{E}}(x)||_n + ||[c_{\mathcal{I}}(x)]^-||_n$$

  and the new objective function $Q_n(x, \mu) = f(x) + \mu h_n(x)$.

- Given $\mu_0 > 0$, tolerance $\tau > 0$, and a starting point x_0^s:

- For $k = 0, 1, 2, \ldots$
  - Find an approximate minimizer x_k of $Q_n(x, \mu_k)$, starting at x_k^s. Note that this step may require the use of unconstrained optimization methods that do not rely on derivative information.
  - If $h_n(x_k) \leq \tau$, stop with approximate solution x_k.
  - Otherwise, choose a new penalty parameter $\mu_{k+1} > \mu_k$, and a new starting point x_{k+1}^s.

A practical ℓ_1 penalty method

- One alternative to using non-differentiable optimization consists on approximating the problem with a simpler model. For example, one can linearize the constraints $c_i(x)$ and replace the original objective function f by a quadratic approximation around an initial point x. The resulting penalty function is

  $$q(p,\mu) = f(x)+\nabla f(x)^T p+\frac{1}{2}p^T W p+\mu\sum_{i\in\mathcal{E}}|c_i(x)+\nabla c_i(x)^T p|+\mu\sum_{i\in\mathcal{I}}[c_i(x)+\nabla c_i(x)^T p]^-,$$

  where W is a symmetric matrix which contains second derivative information about f and c_i.

- By introducing artificial variables r_i, s_i, and t_i, it is possible to reformulate the problem of minimizing q as a smooth quadratic programming problem:

  $$\min_{p,r,s,t} f(x) + \frac{1}{2}p^T W p + \nabla f(x)^T p + \mu\sum_{i\in\mathcal{E}}(r_i+s_i) + \mu\sum_{i\in\mathcal{E}} t_i,$$

  subject to

  $$\nabla c_i(x)^T p + c_i(x) = r_i - s_i, \quad i\in\mathcal{E},$$

  $$\nabla c_i(x)^T p + c_i(x) \geq -t_i, \quad i\in\mathcal{I},$$

  $$r, s, t \geq 0.$$

Session 15

Augmented Lagrangian Methods

Augmented Lagrangian for Equality-Constrained Problems

- Consider again the equality-constrained problem

$$\min_x f(x), \quad \text{subject to} \quad c_i(x) = 0, \ i \in \mathcal{E}.$$

- The augmented Lagrangian function $\mathcal{L}_A(x, \lambda, \mu)$ combines the Lagrangian and the quadratic penalty function. In this case, it is given by

$$\mathcal{L}_A(x, \lambda, \mu) = f(x) - \sum_{i \in \mathcal{E}} \lambda_i c_i(x) + \frac{\mu}{2} \sum_{i \in \mathcal{E}} c_i^2(x).$$

Minimization of the augmented Lagrangian

- Minimization of $\mathcal{L}_A(x, \lambda, \mu)$ must be performed with respect to x and λ.

- One possibility is to use a two-stage algorithm. Given the value of the penalty parameter $\mu_k > 0$, in the first stage, the value of λ is fixed at the current estimate λ^k, and one performs minimization with respect to x.

- Suppose x_k is the approximate minimizer of $\mathcal{L}_A(x, \lambda^k, \mu_k)$, the optimality conditions for unconstrained optimization require that

$$0 \approx \nabla_x \mathcal{L}_A(x_k, \lambda^k, \mu_k) = \nabla f(x_k) - \sum_{i \in \mathcal{E}} \left[\lambda_i^k - \mu_k c_i(x_k)\right] \nabla c_i(x_k).$$

- By comparing with the first optimality condition for the original problem, one can deduct that the optimal Lagrange multipliers $\widehat{\lambda}$ are $\widehat{\lambda}_i \approx \lambda_i^k - \mu_k c_i(x_k)$, for all $i \in \mathcal{E}$.

- This suggests the following formula to update the Lagrange multiplier vector (second stage):

$$\lambda_i^{k+1} = \lambda_i^k - \mu_k c_i(x_k), \quad \text{for all } i \in \mathcal{E}.$$

Augmented Lagrangian algorithm

- Given $\mu_0 > 0$, tolerance $\tau_0 > 0$, and an initial solution x_0^s, λ^0.
- For $k = 0, 1, 2, \ldots$
  - Find an approximate minimizer x_k of $\mathcal{L}_A(x, \lambda^k, \mu_k)$, starting at x_k. Terminate when $||\nabla_x \mathcal{L}_A(x_k, \lambda^k, \mu_k)|| \leq \tau_k$.
  - If the algorithm has converged, stop with x_k as approximate solution.
  - Otherwise, update the Lagrange multipliers: $\lambda_i^{k+1} = \lambda_i^k - \mu_k c_i(x_k)$, for all $i \in \mathcal{E}$.
  - Update the penalty parameter $\mu_{k+1} > \mu_k$, and select tolerance τ_{k+1}.
  - Set the starting point for the next iteration: $x_{k+1}^s = x_k$.

Inequality-constrained problems

- Given a general nonlinear program, one can convert it to a problem with equality and bound constraints by introducing slack variables s_i and replacing the inequalities $c_i(x) \geq 0$, $i \in \mathcal{I}$ by

$$c_i(x) - s_i = 0, \quad s_i \geq 0, \quad \text{for all } i \in \mathcal{I}.$$

  Of course, bound constraints in the original problem need not be transformed.

- This reformulation allows us to write any nonlinear program as follows:

$$\min_{x \in \mathbb{R}^n} f(x), \quad \text{subject to } c_i(x) = 0, i = 1, \ldots, m, \quad l \leq x \leq u,$$

  where the slack variables have been incorporated into the unknown vector x, and l and u denote the lower and upper bounds (some components of l may be set to $-\infty$ and some of u to $+\infty$).

Bound-Constrained Augmented Lagrangian

- The bound-constrained Lagrangian (BCL) incorporates only the equality constraints from the previous formulation into the augmented Lagrangian:

$$\mathcal{L}_A(x, \lambda, \mu) = f(x) - \sum_{i=1}^{m} \lambda_i c_i(x) + \frac{\mu}{2} \sum_{i=1}^{m} c_i^2(x).$$

- The bound constraints are enforced in the subproblem, which has the following form:

$$\min_x \mathcal{L}_A(x, \lambda, \mu), \quad \text{subject to } l \geq x \geq u.$$

  Once this problem has been solved approximately, the multipliers λ and the penalty coefficient μ are updated and the process is repeated.

- One way to solve a nonlinear problem with bound constraints (for fixed λ and μ) consists in approximating the objective function with a quadratic model around the current solution x_k, and using the quadratic gradient projection method to improve the solution.