TMA947 Assignment Exercises 6

Edoardo Mangia

October 2024

Contents

1 HW 6.1 **1**

2 E 5.11 **3**

3 E 6.5 **6**

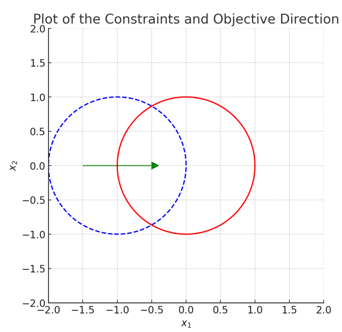
4 E 6.6 **10**

1 HW 6.1

Problem Statement

Consider the optimization problem:

$$\begin{aligned}
\text{Minimize} \quad & -x_1 \quad (6.1) && (1)\\
\text{Subject to} \quad & (x_1+1)^2 + x_2^2 \leq 1 \quad (6.2) && (2)\\
& x_1^2 + x_2^2 \leq 1 \quad (6.3) && (3)
\end{aligned}$$



Lagrangian Relaxation of Constraint (6.2)

Constraint (6.2) is relaxed by introducing a dual variable $\lambda \geq 0$. The Lagrangian $L(x, \lambda)$ is defined as:

$$L(x, \lambda) = -x_1 + \lambda \left((x_1 + 1)^2 + x_2^2 - 1\right)$$

Dual Function $q(\lambda)$

The dual function is defined by:

$$q(\lambda) = \inf\{L(x, \lambda) \mid x_1^2 + x_2^2 \leq 1\}$$

Expanding the Lagrangian:

$$\begin{aligned} q(\lambda) &= \inf\left\{-x_1 + \lambda\left(x_1^2 + 2x_1 + 1 + x_2^2 - 1\right) \mid x_1^2 + x_2^2 \leq 1\right\} \\ &= \inf\left\{\lambda x_1^2 + \lambda x_2^2 + (2\lambda - 1)x_1 \mid x_1^2 + x_2^2 \leq 1\right\} \\ &= \inf\left\{\lambda(x_1^2 + x_2^2) + (2\lambda - 1)x_1 \mid x_1^2 + x_2^2 \leq 1\right\} \end{aligned}$$

Determining the Dual Function $q(\lambda)$

Case 1: $\lambda \geq \frac{1}{4}$

In this case, the stationary point is feasible. Minimizing the Lagrangian yields:

$$q(\lambda) = -\frac{(2\lambda - 1)^2}{4\lambda}$$

Case 2: $\lambda < \frac{1}{4}$

Here, the minimum occurs on the boundary $x_1^2 + x_2^2 = 1$. Evaluating at $x_1 = 1$ gives:

$$q(\lambda) = 3\lambda - 1$$

Summary of the Dual Function

Combining both cases, the dual function is:

$$q(\lambda) = \begin{cases} 3\lambda - 1 & \text{if } \lambda < \frac{1}{4}, \\ -\frac{(2\lambda - 1)^2}{4\lambda} & \text{if } \lambda \geq \frac{1}{4}. \end{cases}$$

Subdifferential of $q(\lambda)$ at $\lambda = \frac{1}{4}$

The subdifferential $\partial q(\lambda)$ at $\lambda = \frac{1}{4}$ consists of all possible subgradients (slopes of supporting lines) at that point.

Left Derivative

Approaching $\lambda = \frac{1}{4}$ from below:

$$\frac{d}{d\lambda}(3\lambda - 1) = 3$$

Right Derivative

Approaching $\lambda = \frac{1}{4}$ from above:

$$\frac{d}{d\lambda}\left(-\frac{(2\lambda - 1)^2}{4\lambda}\right)\bigg|_{\lambda=\frac{1}{4}} = 3$$

Conclusion

Since both the left and right derivatives at $\lambda = \frac{1}{4}$ are equal to 3, the function $q(\lambda)$ is differentiable at this point. Therefore, the subdifferential at $\lambda = \frac{1}{4}$ is the singleton set containing only the value 3.

$$\partial q\left(\frac{1}{4}\right) = \{\, 3 \,\}$$

2 E 5.11

Convex Optimization Problem

Consider the unconstrained optimization problem:

$$\min_{x \in \mathbb{R}^2} f(x)$$

where f is a convex function.

Part (a)

Given:

- Point $\bar{x} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$
- $f(\bar{x}) = 2$
- Subgradient at $\bar{x}$: $g = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

Objective: Determine the optimal value f^* and the minimizer x^*.

Analysis

1. **Subgradient Condition for Optimality:**

   For a convex function f, a point $\bar{x}$ is a minimizer if and only if the zero vector is a subgradient at $\bar{x}$, i.e., $0 \in \partial f(\bar{x})$.

   Here, the given subgradient $g = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \neq 0$, which implies that $\bar{x}$ is **not** a minimizer.

2. **Lower Bound on $f(x)$:**

   Using the subgradient g, for any $x \in \mathbb{R}^2$:

   $$f(x) \geq f(\bar{x}) + g^T(x - \bar{x})$$

   Substituting the given values:

   $$f(x) \geq 2 + \begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 - 2 \\ x_2 - 1 \end{pmatrix} = 2 + (x_1 - 2) - (x_2 - 1) = x_1 - x_2 + 1$$

3. **Implications for the Optimal Value f^*:**

   The inequality $f(x) \geq x_1 - x_2 + 1$ suggests that $f(x)$ is bounded below by the linear function $x_1 - x_2 + 1$. However, since x_1 and x_2 can take any real values, the expression $x_1 - x_2 + 1$ can be made arbitrarily negative by choosing $x_1 - x_2$ sufficiently negative. Therefore, the infimum of $f(x)$ is $-\infty$, i.e., $f^* = -\infty$.

4. **Existence of a Minimizer x^*:**

   Since the infimum of $f(x)$ is $-\infty$, there is **no finite minimizer** x^* in $\mathbb{R}^2$.

Conclusion for Part (a)

$$f^* = -\infty \quad \text{and} \quad \text{there is no minimizer } x^* \text{ in } \mathbb{R}^2.$$

Part (b)

Given:

- Point $\tilde{x} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
- $f(\tilde{x}) = -1$
- Subgradients at $\tilde{x}$:

  $$g_1 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \quad g_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad g_3 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Objective: Determine the optimal value f^* and the minimizer x^*.

Analysis

1. **Subgradient Condition for Optimality:**

   For a convex function f, $\tilde{x}$ is a minimizer if and only if $0 \in \partial f(\tilde{x})$, where $\partial f(\tilde{x})$ is the convex hull of all subgradients at $\tilde{x}$.

2. **Determining if Zero is in the Subgradient Set:**

   It must be checked whether there exist non-negative scalars $\lambda_1, \lambda_2, \lambda_3$ such that:

   $$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

   and

   $$\lambda_1 g_1 + \lambda_2 g_2 + \lambda_3 g_3 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

   Substituting the given subgradients:

   $$\lambda_1 \begin{pmatrix} -1 \\ 0 \end{pmatrix} + \lambda_2 \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \lambda_3 \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

   This yields the system of equations:

   $$-\lambda_1 + \lambda_2 + \lambda_3 = 0 \quad (1)$$

   $$2\lambda_2 - \lambda_3 = 0 \quad (2)$$

   $$\lambda_1 + \lambda_2 + \lambda_3 = 1 \quad (3)$$

3. **Solving the System:**

   From equation (2):

   $$\lambda_3 = 2\lambda_2$$

   Substituting $\lambda_3 = 2\lambda_2$ into equation (1):

   $$-\lambda_1 + \lambda_2 + 2\lambda_2 = 0 \quad \Rightarrow \quad \lambda_1 = 3\lambda_2$$

   Substituting $\lambda_1 = 3\lambda_2$ and $\lambda_3 = 2\lambda_2$ into equation (3):

   $$3\lambda_2 + \lambda_2 + 2\lambda_2 = 6\lambda_2 = 1 \quad \Rightarrow \quad \lambda_2 = \frac{1}{6}$$

   Therefore:

   $$\lambda_1 = \frac{1}{2}, \quad \lambda_3 = \frac{1}{3}$$

   All λ_i are non-negative and sum to 1, satisfying the convex combination condition.

4. **Conclusion on Optimality:**

   Since $0 \in \partial f(\tilde{x})$, $\tilde{x}$ is a **minimizer** of f.

   Optimal Value and Minimizer:

   $$f^* = f(\tilde{x}) = -1$$

   $$x^* = \tilde{x} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Conclusion for Part (b)

$$f^* = -1 \quad \text{and} \quad x^* = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

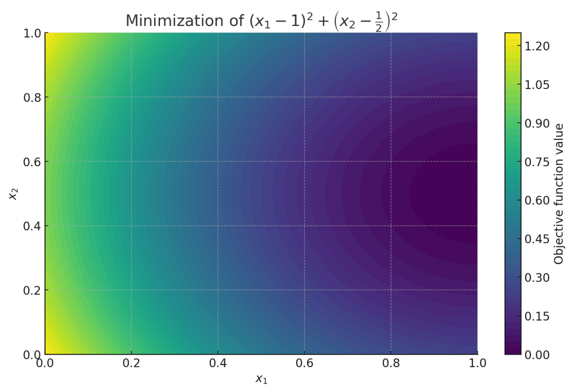
This is established by verifying that the zero vector lies within the subgradient set at $\tilde{x}$, fulfilling the optimality condition for convex optimization.

3 E 6.5

Optimization Problem

Consider the following optimization problem:

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{x}) = (x_1 - 1)^2 + \left(x_2 - \frac{1}{2}\right)^2 \\ \text{Subject to} \quad & 0 \leq x_1 \leq 1, \\ & 0 \leq x_2 \leq 1. \end{aligned}$$



Frank-Wolfe Algorithm Iterates

The Frank-Wolfe algorithm is applied starting from the initial point $\mathbf{x}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. Below are the first five iterates generated by the algorithm.

1. Initial Point ($\mathbf{x}_0$)

$$\mathbf{x}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Gradient at $\mathbf{x}_0$:

$$\nabla f(\mathbf{x}_0) = \begin{pmatrix} -2 \\ -1 \end{pmatrix}$$

Linear Minimization Oracle ($\mathbf{s}_0$):

$$\mathbf{s}_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Step Size (γ_0):

$$\gamma_0 = 1$$

Next Iterate ($\mathbf{x}_1$):

$$\mathbf{x}_1 = (1 - \gamma_0)\mathbf{x}_0 + \gamma_0 \mathbf{s}_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

2. First Iteration ($\mathbf{x}_1$)

$$\mathbf{x}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Gradient at $\mathbf{x}_1$:

$$\nabla f(\mathbf{x}_1) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Linear Minimization Oracle ($\mathbf{s}_1$):

$$\mathbf{s}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Step Size (γ_1):

$$\gamma_1 = \frac{2}{3}$$

Next Iterate ($\mathbf{x}_2$):

$$\mathbf{x}_2 = \frac{1}{3}\mathbf{x}_1 + \frac{2}{3}\mathbf{s}_1 = \begin{pmatrix} \frac{1}{3} \\ \frac{1}{3} \end{pmatrix}$$

3. Second Iteration ($\mathbf{x}_2$)

$$\mathbf{x}_2 = \begin{pmatrix} \frac{1}{3} \\ \frac{1}{3} \end{pmatrix}$$

Gradient at $\mathbf{x}_2$:

$$\nabla f(\mathbf{x}_2) = \begin{pmatrix} -\frac{4}{3} \\ -\frac{1}{3} \end{pmatrix}$$

Linear Minimization Oracle ($\mathbf{s}_2$):

$$\mathbf{s}_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Step Size (γ_2):

$$\gamma_2 = \frac{1}{2}$$

Next Iterate ($\mathbf{x}_3$):

$$\mathbf{x}_3 = \frac{1}{2}\mathbf{x}_2 + \frac{1}{2}\mathbf{s}_2 = \begin{pmatrix} \frac{2}{3} \\ \frac{2}{3} \end{pmatrix}$$

4. Third Iteration ($\mathbf{x}_3$)

$$\mathbf{x}_3 = \begin{pmatrix} \frac{2}{3} \\ \frac{2}{3} \end{pmatrix}$$

Gradient at $\mathbf{x}_3$:

$$\nabla f(\mathbf{x}_3) = \begin{pmatrix} -\frac{2}{3} \\ \frac{1}{3} \end{pmatrix}$$

Linear Minimization Oracle ($\mathbf{s}_3$):

$$\mathbf{s}_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Step Size (γ_3):

$$\gamma_3 = \frac{2}{5}$$

Next Iterate ($\mathbf{x}_4$):

$$\mathbf{x}_4 = \frac{3}{5}\mathbf{x}_3 + \frac{2}{5}\mathbf{s}_3 = \begin{pmatrix} 0.8 \\ 0.4 \end{pmatrix}$$

5. Fourth Iteration ($\mathbf{x}_4$)

$$\mathbf{x}_4 = \begin{pmatrix} 0.8 \\ 0.4 \end{pmatrix}$$

Gradient at $\mathbf{x}_4$:

$$\nabla f(\mathbf{x}_4) = \begin{pmatrix} -0.4 \\ -0.2 \end{pmatrix}$$

Linear Minimization Oracle ($\mathbf{s}_4$):

$$\mathbf{s}_4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Step Size (γ_4):

$$\gamma_4 = \frac{1}{3}$$

Next Iterate ($\mathbf{x}_5$):

$$\mathbf{x}_5 \approx \frac{2}{3}\mathbf{x}_4 + \frac{1}{3}\mathbf{s}_4 = \begin{pmatrix} 0.8667 \\ 0.6 \end{pmatrix}$$

Summary of Iterates

$$
\begin{aligned}
\mathbf{x}_0 &= \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \\
\mathbf{x}_1 &= \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \\
\mathbf{x}_2 &= \begin{pmatrix} \frac{1}{3} \\ \frac{1}{3} \end{pmatrix}, \\
\mathbf{x}_3 &= \begin{pmatrix} \frac{2}{3} \\ \frac{2}{3} \end{pmatrix}, \\
\mathbf{x}_4 &= \begin{pmatrix} 0.8 \\ 0.4 \end{pmatrix}, \\
\mathbf{x}_5 &\approx \begin{pmatrix} 0.8667 \\ 0.6 \end{pmatrix}.
\end{aligned}
$$

Behavior of the Frank-Wolfe Algorithm

1. **Oscillatory Movement Towards Optimum:** The iterates oscillate between boundary points (e.g., $(1, 1)$ and $(1, 0)$) and interior points within the feasible region. This oscillatory behavior is characteristic of the Frank-Wolfe algorithm, especially in problems with multiple constraints.

2. **Convergence Towards the Optimal Solution:** Despite the oscillations, the iterates progressively approach the optimal solution. For this problem, the unconstrained minimum is at $\mathbf{x}^* = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}$. The iterates are converging towards this point while respecting the constraints $0 \le x_1 \le 1$ and $0 \le x_2 \le 1$.

3. **Convex Combination of Extreme Points:** Each new iterate is constructed as a convex combination of the current point and a vertex of the feasible set determined by the linear minimization oracle. This ensures that all iterates remain within the feasible region.

4. **Step Size Influence:** The chosen step sizes γ_k decrease over iterations, allowing the algorithm to make smaller adjustments as it homes in on the optimal solution. This balances exploration of the feasible region with convergence.

5. **Efficiency in Smooth Convex Problems:** For smooth convex problems like the one presented, the Frank-Wolfe algorithm efficiently navigates towards the minimum by leveraging gradient information and the structure of the feasible set.

4 E 6.6

Optimization Problem

Consider the following optimization problem:

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{x}) = (x_1 - 1)^2 + \left(x_2 - \frac{1}{2}\right)^2 \\ \text{Subject to} \quad & 0 \leq x_1 \leq 1, \\ & 0 \leq x_2 \leq 1. \end{aligned}$$

Simplicial Decomposition Algorithm Iterates

The Simplicial Decomposition Algorithm is applied starting from the initial point $\mathbf{x}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. Below are the first five iterates generated by the algorithm.

1. Initial Point ($\mathbf{x}_0$)

$$\mathbf{x}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Gradient at $\mathbf{x}_0$:

$$\nabla f(\mathbf{x}_0) = \begin{pmatrix} -2 \\ -1 \end{pmatrix}$$

Active Set ($\mathcal{S}_0$):

$$\mathcal{S}_0 = \{\mathbf{x}_0\}$$

Search Direction:

$$\mathbf{d}_0 = \mathbf{x}^* - \mathbf{x}_0 = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}$$

Step Size (γ_0):

$$\gamma_0 = 1$$

Next Iterate ($\mathbf{x}_1$):

$$\mathbf{x}_1 = \mathbf{x}_0 + \gamma_0 \mathbf{d}_0 = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}$$

2. First Iteration ($\mathbf{x}_1$)

$$\mathbf{x}_1 = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}$$

Gradient at $\mathbf{x}_1$:

$$\nabla f(\mathbf{x}_1) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Active Set ($\mathcal{S}_1$):

$$\mathcal{S}_1 = \{\mathbf{x}_0, \mathbf{x}_1\}$$

Search Direction:

$$\mathbf{d}_1 = \mathbf{x}^* - \mathbf{x}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Step Size (γ_1):

$$\gamma_1 = 0 \quad \text{(Termination Condition Met)}$$

Next Iterate ($\mathbf{x}_2$):

$$\mathbf{x}_2 = \mathbf{x}_1 = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}$$

3. Second Iteration ($\mathbf{x}_2$)

Since the search direction $\mathbf{d}_1 = \mathbf{0}$, the algorithm has converged to the optimal solution.

$$\mathbf{x}_2 = \mathbf{x}_1 = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}$$

Algorithm Terminates.

Summary of Iterates

$$\begin{aligned} \mathbf{x}_0 &= \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \\ \mathbf{x}_1 &= \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}, \\ \mathbf{x}_2 &= \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}. \end{aligned}$$

Behavior of the Simplicial Decomposition Algorithm

1. **Rapid Convergence:** The algorithm converges to the optimal solution in just two iterations. This is due to the simplicity of the feasible set and the objective function being quadratic and convex.
2. **Building the Active Set:** The active set $\mathcal{S}_k$ grows by adding new points that improve the approximation of the feasible region. In this case, the algorithm identifies the optimal vertex quickly.
3. **Exact Optimality in Few Steps:** For problems where the optimal solution lies at a vertex of the feasible set, the simplicial decomposition algorithm can find the solution in a minimal number of iterations.
4. **Efficient Utilization of Gradient Information:** The algorithm effectively uses gradient information to identify promising directions towards the optimum.

Comparison to Frank-Wolfe Algorithm

1. **Number of Iterations:**
    - **Frank-Wolfe:** Took five iterations to approach the optimal solution.
    - **Simplicial Decomposition:** Converged in two iterations.
2. **Convergence Behavior:**
    - **Frank-Wolfe:** Exhibited oscillatory behavior, alternating between boundary points and interior points, gradually approaching the optimal solution.
    - **Simplicial Decomposition:** Directly identified and moved towards the optimal vertex without oscillation.
3. **Active Set Management:**
    - **Frank-Wolfe:** Does not explicitly maintain an active set of vertices; each iterate is a convex combination based on the current gradient.
    - **Simplicial Decomposition:** Actively builds and updates an active set of vertices that form a simplex approximating the feasible region.
4. **Applicability:**
    - **Frank-Wolfe:** More suitable for large-scale problems where maintaining a simplex is computationally expensive.
    - **Simplicial Decomposition:** Can be more efficient for smaller problems or when rapid convergence is desired.
5. **Solution Path:**
    - **Frank-Wolfe:** Iterates move within the feasible region, often requiring multiple steps to hone in on the optimum.
    - **Simplicial Decomposition:** Quickly converges by selecting vertices that span the simplex towards the optimal solution.