Numerical optimization : theory and applications

Ammar Mian
Associate professor, LISTIC, Université Savoie Mont Blanc

Outline

The Missing Piece: Understanding the Saddle Point Structure

What we covered previously: KKT conditions tell us *what* the solution looks like

What we missed: *How* to optimize the Lagrangian to find this solution

Key Question

Given $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \sum_i \lambda_i c_i(\mathbf{x})$, how do we optimize over $(\mathbf{x}, \boldsymbol{\lambda})$?

The fundamental insight: The KKT conditions emerge from a *saddle point* structure where:

- We **minimize** over primal variables $\mathbf{x}$
- We **maximize** over dual variables $\boldsymbol{\lambda} \geq 0$

This opposite optimization behavior is *not* arbitrary—it emerges naturally from the mathematical structure of constrained optimization.

Why the Minus Sign Creates the Right Incentives

Consider our Lagrangian: $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \sum_i \lambda_i c_i(\mathbf{x})$

What happens if we minimize over both variables?

For inequality constraint $c_i(\mathbf{x}) \geq 0$:

- When $c_i(\mathbf{x}) > 0$ (constraint satisfied with slack)
- Term $-\lambda_i c_i(\mathbf{x})$ becomes more negative as λ_i increases
- Minimizing over λ_i would drive $\lambda_i \to +\infty$, making $\mathcal{L} \to -\infty$
- This creates an unbounded optimization problem

The Resolution

We must **maximize** over $\lambda_i \geq 0$. When $c_i(\mathbf{x}) > 0$, maximization drives $\lambda_i \to 0$ to make $\mathcal{L}$ as large as possible, giving us complementarity: $\lambda_i c_i(\mathbf{x}) = 0$.

The minus sign in the Lagrangian creates the correct incentive structure for the dual variables to encode constraint shadow prices through the saddle point property.

The Saddle Point Property

Theorem (Saddle Point Characterization)

$(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)$ *solves the constrained optimization problem if and only if it is a saddle point of the Lagrangian:*

$$\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}) \leq \mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star) \leq \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^\star)$$

for all feasible $\mathbf{x}$ *and all* $\boldsymbol{\lambda} \geq 0$.

Interpretation:

- Left inequality: $\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda})$ is *maximized* over $\boldsymbol{\lambda}$ at $\boldsymbol{\lambda}^\star$
- Right inequality: $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^\star)$ is *minimized* over $\mathbf{x}$ at $\mathbf{x}^\star$

Economic Insight

Dual variables $\boldsymbol{\lambda}^\star$ represent **shadow prices**—the marginal value of relaxing constraints. Maximization over $\boldsymbol{\lambda}$ finds the economically meaningful constraint valuations.

Illustrative Example: The Saddle Point in Action

Problem: $\min f(x) = -(x-3)^2$ subject to $x \geq 1$
Lagrangian: $\mathcal{L}(x, \lambda) = -(x-3)^2 - \lambda(x-1)$

The conflict: Objective wants $x \to -\infty$, constraint forces $x^\star = 1$

Saddle point analysis:

$$\frac{\partial \mathcal{L}}{\partial x} = -2(x-3) - \lambda = 0 \quad \text{(Stationarity)} \tag{1}$$

$$\text{At } x^\star = 1: \quad -2(1-3) - \lambda = 0 \Rightarrow \lambda^\star = 4 \tag{2}$$

Verification of saddle property:

- Fix $\lambda = 4$: $\mathcal{L}(x, 4) = -(x-3)^2 - 4(x-1)$ has unique minimum at $x = 1$
- Fix $x = 1$: $\mathcal{L}(1, \lambda) = -4$ (constant, satisfying max condition)

Shadow price: $\lambda^\star = 4$ means relaxing $x \geq 1$ to $x \geq 1 - \epsilon$ improves objective by $\approx 4\epsilon$.

From Theory to Algorithm: Projected Gradient Method

The saddle point structure naturally suggests an **alternating optimization** scheme:

Projected Gradient Algorithm

Initialize: $\mathbf{x}^0, \boldsymbol{\lambda}^0 \geq 0$
For $k = 0, 1, 2, \ldots$ **until convergence:**

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^k, \boldsymbol{\lambda}^k) \qquad \text{(Primal descent)}$$

$$\boldsymbol{\lambda}^{k+1} = \max(0, \boldsymbol{\lambda}^k + \beta_k \nabla_{\boldsymbol{\lambda}} \mathcal{L}(\mathbf{x}^{k+1}, \boldsymbol{\lambda}^k)) \qquad \text{(Dual ascent)}$$

Key components:

- **Primal step**: Gradient descent on $\mathcal{L}$ with respect to $\mathbf{x}$
- **Dual step**: Projected gradient ascent on $\mathcal{L}$ with respect to $\boldsymbol{\lambda}$
- **Projection**: $\max(0, \cdot)$ ensures dual feasibility $\boldsymbol{\lambda} \geq 0$

Understanding the Gradient Components

For our general Lagrangian $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \sum_i \lambda_i c_i(\mathbf{x})$:

Primal gradient:

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \nabla f(\mathbf{x}) - \sum_i \lambda_i \nabla c_i(\mathbf{x})$$

Dual gradient:

$$\frac{\partial \mathcal{L}}{\partial \lambda_i} = -c_i(\mathbf{x})$$

Algorithm Updates

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \left(\nabla f(\mathbf{x}^k) - \sum_i \lambda_i^k \nabla c_i(\mathbf{x}^k) \right) \tag{3}$$

$$\lambda_i^{k+1} = \max(0, \lambda_i^k + \beta_k c_i(\mathbf{x}^{k+1})) \quad \forall i \tag{4}$$

Intuition: Dual variables increase when constraints are violated ($c_i < 0$) and decrease when constraints have slack ($c_i > 0$), naturally driving toward complementarity

Algorithm Implementation for Our Exercise

Recall our problem:

$$\begin{aligned} \text{minimize} \quad & f(x, y) = (x-2)^2 + (y-2)^2 && (5) \\ \text{subject to:} \quad & g(x, y) = x + y - 2 = 0 && (6) \\ & h_1(x, y) = x \geq 0 && (7) \\ & h_2(x, y) = y \geq 0 && (8) \end{aligned}$$

Lagrangian:

$$\mathcal{L}(x, y, \lambda, \mu_1, \mu_2) = (x-2)^2 + (y-2)^2 - \lambda(x + y - 2) - \mu_1 x - \mu_2 y$$

Gradients:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial x} &= 2(x-2) - \lambda - \mu_1 && (9) \\ \frac{\partial \mathcal{L}}{\partial y} &= 2(y-2) - \lambda - \mu_2 && (10) \\ \frac{\partial \mathcal{L}}{\partial \lambda} &= -(x + y - 2) && (11) \\ \frac{\partial \mathcal{L}}{\partial \mu_1} &= -x, \quad \frac{\partial \mathcal{L}}{\partial \mu_2} = -y && (12) \end{aligned}$$

Projected Gradient Steps for Our Exercise

Algorithm updates:

$$x^{k+1} = x^k - \alpha(2(x^k - 2) - \lambda^k - \mu_1^k) \tag{13}$$

$$y^{k+1} = y^k - \alpha(2(y^k - 2) - \lambda^k - \mu_2^k) \tag{14}$$

$$\lambda^{k+1} = \lambda^k + \beta(x^{k+1} + y^{k+1} - 2) \tag{15}$$

$$\mu_1^{k+1} = \max(0, \mu_1^k - \beta x^{k+1}) \tag{16}$$

$$\mu_2^{k+1} = \max(0, \mu_2^k - \beta y^{k+1}) \tag{17}$$

Expected convergence: $(x^\star, y^\star) = (1, 1)$ with $\lambda^\star = -2$, $\mu_1^\star = \mu_2^\star = 0$

Key Insight

The inequality constraints $x \geq 0, y \geq 0$ are **inactive** at the solution because the optimal point $(1, 1)$ lies in the interior of the feasible region. Therefore $\mu_1^\star = \mu_2^\star = 0$ by complementarity.

Corrected Implementation and Key Takeaways

Implementation insight: The projected gradient method will automatically handle the constraint activity determination through the projection steps.

Algorithm behavior:

- Algorithm starts with some initial guess
- Primal variables evolve toward $(1, 1)$ due to objective function pull
- Dual variables for inactive constraints get projected to zero
- Equality constraint multiplier converges to $\lambda^\star = -2$

Main Learning Objectives

1. Saddle point structure emerges from constraint-objective conflicts
2. Opposite optimization directions (min over $\mathbf{x}$, max over $\boldsymbol{\lambda}, \boldsymbol{\mu}$) are mathematically necessary
3. Projected gradient algorithm implements this structure computationally
4. Shadow prices have economic meaning: $\lambda^\star = -2$ means relaxing the constraint worsens the objective