Optimization

Lecture 8: Deconstraining

Parin Chaipunya
KMUTT
└ Mathematics @ Faculty of Science
└ The Joint Graduate School of Energy and Environments

Areas of research:

- Multi-agent optimization: Bilevel programs, Game theory
- Optimization modeling: mainly focused on energy and environmental applications

parin.cha@kmutt.ac.th
parinchaipunya.com
github.com/parinchaipunya

Last update: Januaray 2026

Table of contents

Scopes

The simple goal of these slides is present different approaches to transform (or relax) a constrained optimization problem into an unconstrained one.

Section 1

Direct substitution

Direct substitution

In some problems, especially those with only equality constraints, we could substitute directly the constraints into the objective function which effectively and completely **removes the constraints**.

However, as we shall see right after, this approach is narrowly usable in practical problems.

Direct substitution

The following is a typical situation where we could use the direct substitution technique

$$\begin{cases} \max_{x,y} & xy \\ \text{s.t.} & x + 2y = 100\,. \end{cases}$$

One could simply substitute, using the constraint, that $x = 100 - 2y$ into the objective function. The optimization problem them becomes

$$\max_{y} \quad (100 - 2y)y\,,$$

which is unconstrained.

Direct substitution

Let us add an additional difficulty to the problem to make it more realistic.

$$\begin{cases} \max\limits_{x,y} & xy \\ \text{s.t.} & x + 2y = 100 \\ & x, y \geq 0\,. \end{cases}$$

From the equality constraint, we still have $x = 100 - 2y$. However, to respect the constraint $x \geq 0$, we need $y \leq 50$.
The problem is now reduced to

$$\begin{cases} \max\limits_{y} & J(y) = (100 - 2y)y \\ \text{s.t.} & 0 \leq y \leq 50\,. \end{cases}$$

Direct substitution

To further reduce it to an unconstrained problem, we need to take into account that

- $J(y) > 0$ for some $0 \leq x \leq 50$. This means

$$\max_{y \in \mathbb{R}} J(y) \geq \max_{0 \leq y \leq 50} J(y) > 0 \tag{1}$$

- $J(y) < 0$ for all $y < 0$ and all $y > 50$. Complementing (1), we obtain

$$\max_{y \in \mathbb{R}} J(y) \geq \max_{0 \leq y \leq 50} J(y) > 0 > J(z) \qquad \forall z \notin [0, 50] \, .$$

This means the unconstrained maximizers of J would not occur outside of the constraint interval $[0, 50]$ — there is no need to keep the constraint.

Finally we arrive at an equivalent unconstrained problem

$$\max_{y} \quad (100 - 2y)y \, .$$

Direct substitution

To conclude, the direct substitution is not widely usable in practice as one could notice that it requires a lot of manual intervention and analysis even in a very simple case.
A bottom line is that we require a more systematic method to transform a constrained problem into an unconstrained one.

Section 2

Penelty method

Penalty method

Consider the problem

$$\begin{cases} \min \quad J(x) & (2a) \\ \text{s.t.} \quad g_j(x) \leq 0 \quad j = 1, \ldots, r & (2b) \\ \qquad\;\; h_k(x) = 0 \quad k = 1, \ldots, \ell\,. & (2c) \end{cases}$$

In the **penalty method**, instead of handling the problem (2) using constrained optimization, we

1. remove all the constraints, and
2. add penalization terms for the constraints that are not satisfied.

Penalized problem

We call a function $Q : \mathbb{R} \to \mathbb{R}_+$ a **penalization function** if

- $Q(0) = 0$, and
- $Q(t) > 0$ for all $t \neq 0$.

The penalized problem now reads

$$\min \quad J(x) + \rho \left(\sum_{j=1}^{r} Q_j^{\leq}\Big(\max\{0, g_j(x)\} \Big) + \sum_{k=1}^{\ell} Q_k^{=}\Big(h_k(x) \Big) \right), \tag{3}$$

where $Q_j^{\leq}$'s and $Q_k^{=}$'s are penalization functions, and $\rho > 0$ is the weight parameter.

By putting menalization terms, the solution does not favor a point x that produces high penalty and try to stay within the constraint set. The constraint is respected more with a higher parameter ρ.

One should expect that **the solution of** (3) **converges to that of the original problem** (2) **as** $\rho \to +\infty$.

However, it is important to note that **a solution of the penalized problem may violate constraints**.

Penalized problem

Two common penalization functions are

$$Q(t) = t^2 \quad \text{and} \quad Q(t) = |t| \, .$$

Then, one may arrive at the following variants of penalized problems:

$$\min \quad J(x) + \rho \left(\sum_{j=1}^{r} \max\{0, g_j(x)\} + \sum_{k=1}^{\ell} \Big(h_k(x) \Big)^2 \right) ,$$

or more commonly,

$$\min \quad \widehat{J}_\rho(x) = J(x) + \rho \left(\sum_{j=1}^{r} \Big(\max\{0, g_j(x)\} \Big)^2 + \sum_{k=1}^{\ell} \Big(h_k(x) \Big)^2 \right) .$$

In the latter one, we have

$$\nabla \widehat{J}_\rho(x) = \nabla J(x) + 2\rho \left(\sum_{j=1}^{r} \max\{0, g_j(x)\} \nabla g_j(x) + \sum_{k=1}^{\ell} h_k(x) \nabla h_k(x) \right) .$$

Section 3

Barrier method

Problem scope

We consider the following problem, which only has inequality constraints.

$$\begin{cases} \min \quad J(x) & \text{(4a)} \\ \text{s.t.} \quad g_j(x) \leq 0 \quad j = 1, \ldots, r\,. & \text{(4b)} \end{cases}$$

Barrier method

- The **barrier method** is somewhat similar to the penalty method in that it adds extra terms to the objective function.
- The difference is that, in the barrier method, we add terms that **penalize any point** x **that gets too close to the boundary of the constraint set**.
- This way, it is ensured that the solution **never leaves the feasible region**.
- The con of this method is obvious — it prevents one from reaching the solutions that lie at the boundary of the feasible region.
- Another con is that the barrier method is **not suitable to equality constraints**.

Barrier method

A function $B : \mathbb{R}^- \to \mathbb{R}$ is called a **barrier function** if

$$B(t) \to +\infty \quad \text{as } t \to 0^- \,.$$

The problem (4) is then transformed into

$$\min \quad \widetilde{J}_\rho(x) = J(x) + \rho \sum_{j=1}^{r} B(g_j(x))\,, \tag{5}$$

where $\rho > 0$ is the weight parameter.

Note that $\nabla \widetilde{J}_\rho(x) = \nabla J(x) + \rho \sum_{j=1}^{r} B'(g_j(x)) \nabla g_j(x)$.

One should expect that **the solution of** (5) **converges to that of the original problem** (4) **as** $\rho \to 0^+$.

Log-barrier function

A typical example is the **log-barrier function**

$$B(t) = -\log(-t)\,.$$

In this case, we have the following problem to solve

$$\min \quad \widetilde{J}_\rho(x) = J(x) - \rho \sum_{j=1}^{r} \log(-g_j(x))\,,$$

and we get

$$\nabla \widetilde{J}_\rho(x) = \nabla J(x) - \rho \sum_{j=1}^{r} \frac{\nabla g_j(x)}{g_j(x)}\,.$$

Section 4

Examples

Subsection 1

Toy examples

Example

Let us illustrate the ideas of both penalty and barrier methods by first considering a very simple example.

Example 1

Consider the following optimization problem

$$\begin{cases} \min & x^2 & (6) \\ \text{s.t.} & x \geq 1\,. & (7) \end{cases}$$

(a) Solve (6) using both penalty and barrier methods and compare both results with the exact minimizer.
(b) Let $\rho \to \infty$ in the penalty method and observe the results.
(c) Let $\rho \to 0^+$ in the barrier method and observe the results.

Example

Let us illustrate the ideas of both penalty and barrier methods by first considering a very simple example.

Example 2

Consider the following optimization problem

$$\begin{cases} \min & x^2 + y^2 \qquad (8) \\ \text{s.t.} & x + y \geq 1\,. \qquad (9) \end{cases}$$

(a) Solve (6) using both penalty and barrier methods and compare both results with the exact minimizer.
(b) Let $\rho \to \infty$ in the penalty method and observe the results.
(c) Let $\rho \to 0^+$ in the barrier method and observe the results.

Example

Let us illustrate the ideas of both penalty and barrier methods by first considering a very simple example.

Example 3

Consider the following optimization problem

$$\begin{cases} \min & x^2 + y^2 \quad (10) \\ \text{s.t.} & x + y = 1 \,. \quad (11) \end{cases}$$

(a) Solve (6) using the penalty methods and compare the result with the exact minimizer.
(b) Let $\rho \to \infty$ in the penalty method and observe the results.

Subsection 2

Image denoising

Image denoising

Let us formulate **image denoising** as a constrained optimization problem.
Suppose that a noisy observed grayscale image is $Y \in [0,1]^{m\times n}$. The goal is to find an unknown true image $X \in [0,1]^{m\times n}$ without noise.

The optimiztion problem of image denoising is formulated as

$$\begin{cases} \min\limits_{X} & \|X - Y\|_F^2 + \tau \left[\sum_{i=1}^{m-1} \sum_{j=1}^{n} (X_{i+1,j} - X_{i,j})^2 + \sum_{i=1}^{m} \sum_{j=1}^{n-1} (X_{i,j+1} - X_{i,j})^2 \right] \\ \text{s.t.} & 0 \le X_{i,j} \le 1 \quad \text{for all } i,j\,, \end{cases}$$

where $\tau > 0$ is the weight parameter.

Note that the Frobenius norm of a matrix $U \in \mathbb{R}^{m\times n}$ is $\|U\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} U_{i,j}^2}\,$.

Vectorization

Let us put

$$D_m = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & -1 \end{bmatrix}_{(m-1)\times m} \quad \text{and} \quad D_n = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & -1 \end{bmatrix}_{(n-1)\times n} .$$

Then

$$\sum_{i=1}^{m-1} \sum_{j=1}^{n} (X_{i+1,j} - X_{i,j})^2 = \|D_m X\|_F^2 \quad \text{and} \quad \sum_{i=1}^{m} \sum_{j=1}^{n-1} (X_{i,j+1} - X_{i,j})^2 = \|X {D_n}^t\|_F^2 .$$

Vectorization

Finally, the problem becomes

$$\begin{cases} \min\limits_{X} & J(X) = \|X - Y\|_F^2 + \tau\left[\|D_m X\|_F^2 + \|X {D_n}^t\|_F^2\right] & (12a) \\ \text{s.t.} & 0 \le X_{i,j} \le 1 \quad \forall i = 1, \ldots, m,\ \forall j = 1, \ldots, n\,. & (12b) \end{cases}$$

Also note that

$$\nabla J(X) = 2\Big(X - Y + \tau({D_m}^t D_m X + X {D_n}^t D_n)\Big)\,.$$

Example 4

Test denoising an image by solving (12) using the barrier method.

That's it!

Key takeaways.

- The implementation, pros and cons, and differences between the penalty and barrier methods.

Thank you.