

Optimization in industries

through examples

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

A coordinated course with Betagro

26–27 June 2025

 parin.cha@kmutt.ac.th

 parinchaipunya.com

 github.com/parinchaipunya

Overview

The main objective is to get you to know the **frameworks of an optimization problem**.

We will learn about

- **components** in an optimization problem
 - ▷ optimization sense (**min** and **max**)
 - ▷ objective function
 - ▷ constraints of different types
- **classes** of optimization problems
 - ▷ linear programs
 - ▷ nonlinear programs
 - ◇ quadratic programs
 - ◇ etc.

Resources.

 https://parinchaipunya.com/_posts/optim

 <https://kmutt.me/betagro2025>

Table of contents

General ideas about optimization

Linear programs: LPs

Nonlinear programs: NLPs

Teaser problem: Design of Experiments and Optimization

General ideas about optimization

What is optimization ?

Optimization refers to the process that one **makes a decision** to achieve the **best outcome**.

Where is optimization in the map of data analytics?

The four types of data analytics:

- **Descriptive:** What happened in the past ?
Classical statistical tools, Data visualization, Clustering, etc.
 - **Diagnostic:** Why something happened in the past ?
Data discovery, Data mining, Root cause analysis, etc.
 - **Predictive:** What is likely to happen in the future ?
Regression, Classification, etc.
 - **Prescriptive:** How to make something happen ?
Optimization, Operations research, Reinforcement learning (repeated decision), etc.
- ! One should notice that Optimization remains the main tool when it comes to prescriptive analytics. Machine learning tools are largely the prescriptive and descriptive in nature.
- ! ML tools are also useful tool to incorporate with Optimization.

An **optimization problem** (or a **mathematical program**, or a **mathematical programming problem**) takes the following form.

$$\left\{ \begin{array}{ll} \min / \max & f(x) \\ \text{s.t.} & g_i(x) \leq 0 \quad i = 1, \dots, r \\ & h_j(x) = 0 \quad j = 1, \dots, \ell \\ & x \in C. \end{array} \right.$$

Usually, x is a vector $x = (x_1, \dots, x_n) \in \mathbb{R}^n$.

General framework

An **optimization problem** (or a **mathematical program**, or a **mathematical programming problem**) takes the following form.

$$\left\{ \begin{array}{ll} \text{optimization sense} & \text{objective function} \\ \min / \max & \overbrace{f(x)} \\ \text{s.t.} & \left. \begin{array}{l} g_i(x) \leq 0 \quad i = 1, \dots, r \\ h_j(x) = 0 \quad j = 1, \dots, \ell \\ x \in C \end{array} \right\} \text{constraints} \end{array} \right.$$

What could it be ? — Manufacturing

$$\left\{ \begin{array}{ll} \min & \text{Cost} \\ \text{s.t.} & \text{Demand} \\ & \text{Manufacturing power} \\ & \text{Material limit} \end{array} \right.$$

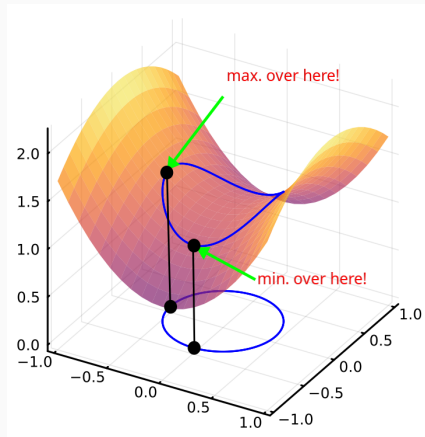
What could it be ? — Resources

$$\left\{ \begin{array}{ll} \min & \text{Resources} \\ \text{s.t.} & \text{Order quantity} \\ & \text{Product quality} \end{array} \right.$$

What could it be ? — Delivery

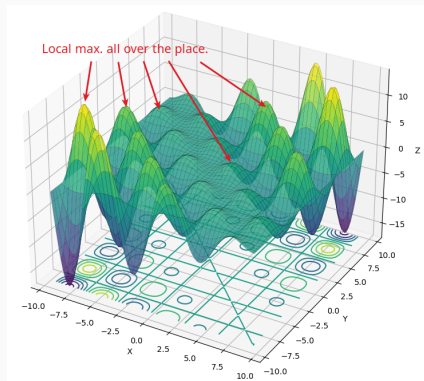
$$\left\{ \begin{array}{ll} \min & \text{Distance travelled} \\ \text{s.t.} & \text{Visits all customers} \\ & \text{No subtour} \end{array} \right.$$
$$\left\{ \begin{array}{ll} \min & \text{Travel time} \\ \text{s.t.} & \text{Visits all customers} \\ & \text{No subtour} \end{array} \right.$$

Optimization problem (with constraints)



We would like to find the **lowest** or **highest** point on the graph **within the feasible set**.

Local vs global optima



In practice, most algorithms stuck at local solutions and it is often difficult to achieve a global one.

Luckily, we can safely reach a global solution for “nice” problems.

Conceptual checkpoint

Let us check our understanding with the following simple example.

Example

Find a solution of the following optimization problems.

$$\begin{cases} \max & x + y \\ \text{s.t.} & 0 \leq x, y \leq 1 \end{cases}$$

$$\begin{cases} \max & x - y \\ \text{s.t.} & 0 \leq x, y \leq 1 \end{cases}$$

$$\begin{cases} \min & x - y \\ \text{s.t.} & x + y = 1 \\ & x, y \geq 0 \end{cases}$$

$$\begin{cases} \min & x + 2y - z \\ \text{s.t.} & x + y = 1 \\ & 3x - 2z = 2 \\ & x, y \geq 0 \\ & z \in \{-1, 1\} \end{cases}$$

Linear programs: LPs

Linear programs

A **linear optimization** (or more commonly called a **linear program: LP**) takes the form

$$\left\{ \begin{array}{ll} \min / \max & e_1x_1 + e_2x_2 + \cdots + e_nx_n + e_0 \\ \text{s.t.} & a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i \quad i = 1, \dots, r \\ & c_{j1}x_1 + c_{j2}x_2 + \cdots + c_{jn}x_n = d_j \quad j = 1, \dots, \ell. \end{array} \right.$$

In vector-matrix form, we write

$$\left\{ \begin{array}{ll} \min / \max & e^\top x \\ \text{s.t.} & Ax \leq b \\ & Cx = d. \end{array} \right.$$

Linear programs (MILP)

A **mixed-integer linear program**: **MILP** is a linear program with an additional constraint that some (possibly all*) of x_i 's are restricted to integers.

$$\left\{ \begin{array}{ll} \min / \max & e_1x_1 + e_2x_2 + \cdots + e_nx_n + e_0 \\ \text{s.t.} & a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i \quad i = 1, \dots, r \\ & c_{j1}x_1 + c_{j2}x_2 + \cdots + c_{jn}x_n = d_j \quad j = 1, \dots, \ell \\ & x_1, \dots, x_k \in \mathbb{Z}. \end{array} \right.$$

The vector-matrix form reads

$$\left\{ \begin{array}{ll} \min / \max & e^T x + e_0 \\ \text{s.t.} & Ax \leq b \\ & Cx = d \\ & x \in \mathbb{Z}^k \times \mathbb{R}^{n-k}. \end{array} \right.$$

*If all variables are integers, one may prefer to call it an **integer linear program**: **ILP**.

Pros and Cons of (M)LPs

Pros:

- simple, intuitive and practical
- guaranteed global optimum
- well-studied
- easy sensitivity analysis
- plenty of solvers and tools.

Cons:

- limitation due to linearity
- computationally slow due to integrality

A practical note:

To overcome the slow computations, **heuristic methods** are used. These methods effectively **improve the objective value**, but there is **no guarantee for optimality**.

Example: When a heuristic is used in a cost minimization problem, one could only expect a **lower cost** but not the **lowest**.

Some (M)LP solvers

Solver	Comments
CPLEX	Commercial. Free for small problems. Fully free for academics.
Gurobi	Commercial. Free for small problems. Fully free for academics.
GAMS	Commercial. Free trial. Fully free for academics.
SCIP [†]	Free and open-source. Also works with nonlinear problems.
HiGHS	Free and open-source. Only for LPs.
CBC	Free and open-source.

[†]We shall use SCIP in our demo problems.

`1_getting_started_pt1.ipynb`

`1_getting_started_pt2.ipynb`

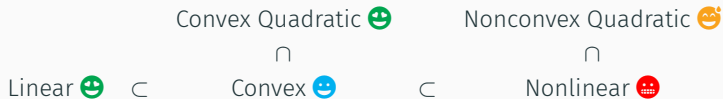
`2_production_and_inventory.ipynb`

`3_blending_peanut_butter.ipynb`

`4_batch_size_multi_product.ipynb`

Nonlinear programs: NLPs

The classes of optimization problems



Convex quadratic programs

A **convex quadratic program**: **convex QP** is an optimization problem taking the form

$$\begin{cases} \min / \max & x^\top E x + e^\top x + e_0 \\ \text{s.t.} & Ax \leq b \\ & Cx = d. \end{cases}$$

We could also formulate the **mixed-integer** version, **MIQP**, as

$$\begin{cases} \min / \max & x^\top E x + e^\top x + e_0 \\ \text{s.t.} & Ax \leq b \\ & Cx = d \\ & x \in \mathbb{Z}^k \times \mathbb{R}^{n-k}. \end{cases}$$

! Note that we still require the constraints to be all linear.

The following use cases are examples of a **convex QP**:

- **All previous problems** but the cost is quadratically estimated.
 - ▷ When the cost/unit increases with the amount or time, a good candidate function is usually a quadratic one.
- **Investment problems** (Markowitz approach).
 - ▷ Expected return vs Variance (risk).
- **Least-squares problems**.
 - ▷ Parameter estimation.
 - ▷ Machine learning problems (Classification/ Regression).
 - ▷ Tracking a target value.
- **Penalization/ regularization techniques** to avoid some undesirable solutions.
 - ▷ Ridge regression, which helps reduce the bias in a regression model.
 - ▷ Penalization of unmet demand.

Not all solvers support nonlinear problems. One should consult the user's manual of the chosen software to see the details.

For your peace of mind, **CPLEX**, **Gurobi** and **SCIP** supports **QPs** (convex and non-convex).

Teaser problem: Design of Experiments and Optimization

Section overview

In this section, we demonstrate **integrating different tools** in analytics that finally **helps you make a better decision**.

The steps are as follows.

Diagnostic step: Use the **DoE** model to help understand the relationships between different factors in a process.

Predictive step: To estimate a **response surface** from the DoE model, which predicts the outcome when the factors are varied.

Prescriptive step: To achieve the **best outcome** by optimizing the model constructed from the response surface.

What is Design of Experiments (DoE) ?

The **Design of experiments: DoE** is a descriptive+diagnostic analytics that helps understanding the effects of different factors in the outcome of a process.

The brutal way to understand this is to try out **all possible combinations** of factors, but this is often **too expensive**.

There are many[‡] techniques in DoE that helps reduce the number of **runs**.

[‡]So many that it constitutes a whole course.

Examples

Example

One wants to observe how the three factors x_1 , x_2 and x_3 affects the outcome of a production process y . Assume that each factor is operated in two modes, either “High” (+1) or “Low” (−1). The following are two examples of experiment tables that could be used.

Table 1: Full factorial DoE.

Run	Factor levels			Outcome
	x_1	x_2	x_3	
1	+1	+1	+1	y_1
2	+1	+1	−1	y_2
3	+1	−1	+1	y_3
4	+1	−1	−1	y_4
5	−1	+1	+1	y_5
6	−1	+1	−1	y_6
7	−1	−1	+1	y_7
8	−1	−1	−1	y_8

Table 2: Fractional factorial DoE.

Run	Factor levels			Outcome
	x_1	x_2	$x_3 = x_1 \cdot x_2$	
1	+1	+1	+1	\hat{y}_1
2	+1	−1	−1	\hat{y}_2
3	−1	+1	−1	\hat{y}_3
4	−1	−1	+1	\hat{y}_4

- designed controlled factors
- collected data

After obtaining the table, one may do further analysis:

- Observe the trends of each factors.
- Screen for the significant factors and drop the insignificant ones.
- If the test is insufficient, then consider adding more runs.

Response Surface Method (RSM)

DoE gives a **diagnostic understanding** of how different factors affect the outcome. It is often required to have implications to the unobserved combination of factors.[§]

Let's consider the **response surface method: RSM**, which is a way to construct an empirical model that helps **predict the outcomes that are not in the DoE table**.

The idea is to **do regression from the DoE table**. It is common to fit the data to a **quadratic surface** (known as the **response surface**), with the assumption that

$$y_i \approx S(x_1, \dots, x_n) \quad S \sim \text{quadratic.}$$

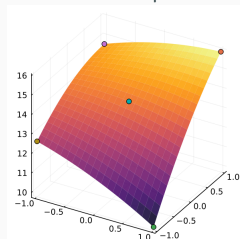


Figure 1: Response surface with 4 corners and one midpoint.

[§]We skipped the screening step.

Response Surface Method (RSM)

For two factors, S has the form

$$S(x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2.$$

For three factors, S has the form

$$S(x_1, x_2, x_3) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{33} x_3^2 + \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{23} x_2 x_3.$$

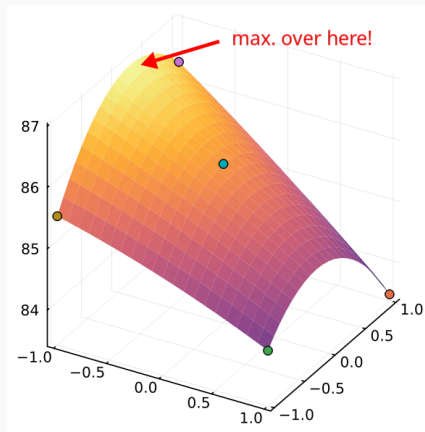
In general when n factors are considered, we need to estimate the coefficients

- β_0 for the intercept,
- β_i for the linear term x_i ,
- β_{ij} for the quadratic term $x_i x_j$.

Regression target: Estimate the coefficients so that the least-squares error is minimized.

Optimization of factors in a process

To achieve the **best outcome**, we need to **optimally select** the right values for each variable. This could be done by **applying optimization to RSM**.



5_doe_pdi_simplified.ipynb

6_doe_pdi_btgdatt.ipynb

Conclusion

We have seen:

- Classes of optimization problems.
- Process-driven + Data-driven optimization modeling.
- Combining different tools in analytics.

Difficulties we have seen:

- Model design choice.
- Lack of data.
- Lack of ability to model a process.
- False inference.
- Nonlinearities.



😊 Thank you for your attendance.

Ayutthaya, 2025
parin.cha@kmutt.ac.th