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

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 reseach, 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.

General framework

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

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

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

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

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

What could it be ? — Manufacturing

$$\begin{cases} \min & \text{Cost} \\ \text{s.t.} & \text{Demand} \\ & \text{Manufacturing power} \\ & \text{Material limit} \end{cases}$$

What could it be ? — Resources

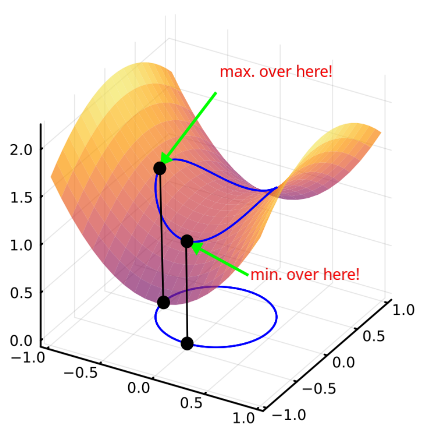
$$\begin{cases} \min & \text{Resources} \\ \text{s.t.} & \text{Order quantity} \\ & \text{Product quality} \end{cases}$$

What could it be ? — Delivery

$$\begin{cases} \min & \text{Distance travelled} \\ \text{s.t.} & \text{Visits all customers} \\ & \text{No subtour} \end{cases}$$

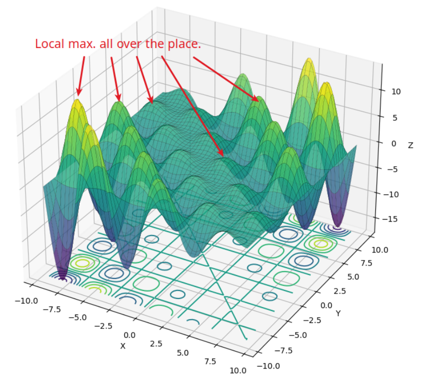
$$\begin{cases} \min & \text{Travel time} \\ \text{s.t.} & \text{Visits all customers} \\ & \text{No subtour} \end{cases}$$

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.

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

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

Linear programs: LPs

Linear programs

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

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

In vector-matrix form, we write

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

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.

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

The vector-matrix form reads

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

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

Pros and Cons of (MI)LPs

Pros:

- simple, intuitive and practical
- guarateed 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 (MI)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.

Demo: Production and inventory management

`production_and_inventory.ipynb`

Demo: Logistics — Lorries and containers

`lorries_and_containers.ipynb`

Nonlinear programs: NLPs

		Convex Quadratic 😃		Nonconvex Quadratic 😅
		∩		∩
Linear 😃	⊂	Convex 😃	⊂	Nonlinear 😬

Convex quadratic programs

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

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

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

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

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

Use cases

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.

Solvers

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 helps 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 constitues 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 x_1	x_2	x_3	Outcome
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 x_1	x_2	$x_3 = x_1 \cdot x_2$	Outcome
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

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

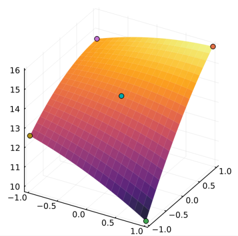


Figure 1: Response surface with 4 corners and one midpoint.

§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

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

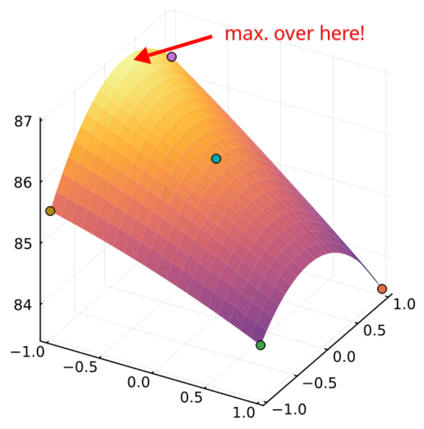
In general when n factors are considered, we need to estimate the coefficients

- 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**.



Demo: DoE and Optimization

`doe_and_optim_1.ipynb`

`doe_and_optim_2.ipynb`

Conclusion

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.
- Data acquisition.
- Lack of ability to model a process.
- False implications.

☻ Thank you for your attendance.

Ayutthaya, 2025

parin.cha@kmutt.ac.th