Engineering Optimization:
Introduction

Yanjie Zhou

https://ieyjzhou.github.io/lab/
Computer and Industry Engineering Group
School of Management
Zhengzhou University

Spring 2023

郑州大学
ZHENGZHOU UNIVERSITY

Outline

Course Information
- Basic information
- Prerequisites
- Coursework

Introduction
- Engineering project
- Decision problems
- Optimization problems
- Engineering optimization
- Design optimization

A case study of engineering optimization problem

Getting started with python
- Geometric interpretation of linear programs
- Python-MIP
- Python tutorials

Homework

Course information

Basic information:

- Name: Engineering Optimization
- ID: 073502.01
- Location: 环 _301(Wednesday); 化 _106(Friday)
- Lectures: 10:10-11:50 (Wednesday, odd week); 14:00-15:40 (Friday, every week);

Instructor:

- Name: Yanjie Zhou
- Email: ieyjzhou@zzu.edu.cn
- Homepage: https://ieyjzhou.github.io/

Prerequisites

You needs to learn the following courses before attending engineering optimization course.

- Calculus I and II
- Linear algebra
- Operation research
- Data structure
- Programming language (familiar with one programming language): C/C++; Python; Java; Matlab; Julia

Coursework

Homework (50%):

- This course has 5 homework.

Final project (50%):

- This course has one project.
- Each team has 3-6 students.

Final exam (0%):

- This course does not has any exam.

Topics I

- Introduction to engineering optimization
  - Finding a optimization problem from industry
  - Decision problem or optimization problem
  - Process for solving an engineering optimization
  - Complexity of a engineering optimization problem
- Mathematical formulations
  - Mathematical definitions
  - Software solvers
  - Linearization methods
- Linear programming
  - Polyhedra and polytopes
  - Optimality conditions
  - Simplex method
  - Interior point method

Topics II

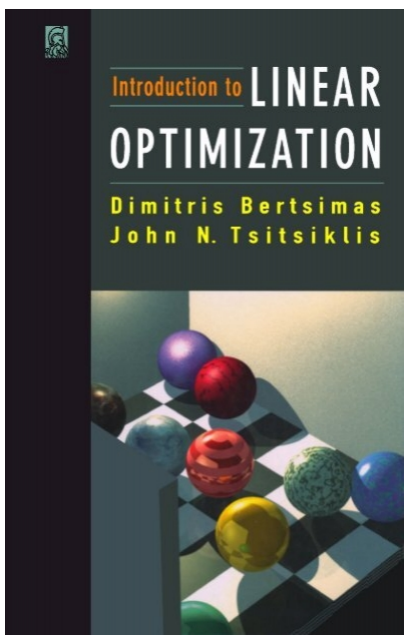
- Integer programming
  - Total unimodularity
  - Branch and bound
  - Branch and cut
- Nonlinear programming
  - Optimality conditions
  - Unconstrained optimization
  - Constrained optimization
  - Local search methods
  - Penalty methods
- Special optimization topics
  - Optimization under uncertainty
  - Multiobjective optimization
  - Linear bilevel optimization
  - Simulation based optimization

Topics III

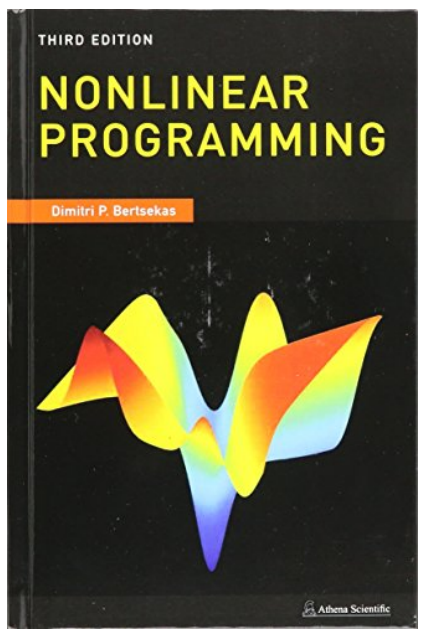
- Nash equilibrium
- Heuristic optimization method
  - Single solution based heuristics
    - Simulated annealing
    - Iterated local search
    - Variable neighborhood search
  - Population based heuristics
    - Genetic algorithms
    - Particle swarm optimization
    - Harmony search algorithm
- Reinforcement Learning
  - Introduction
  - RL for solving combinatorial problem

Textbook I

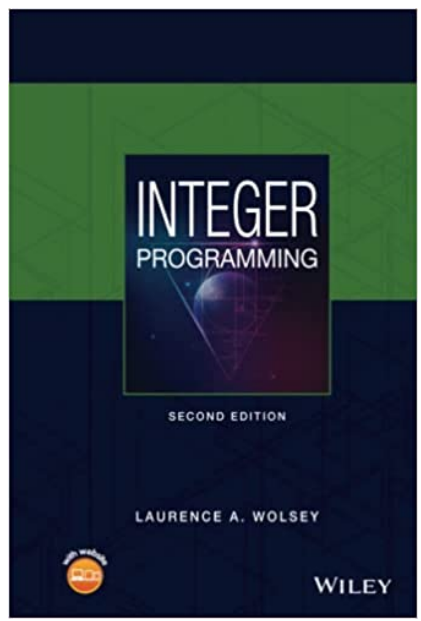
1. Belegundu, A. D., & Chandrupatla, T. R. (2019). Optimization concepts and applications in engineering. Cambridge University Press.
2. Bertsekas, D. P. (2016). Nonlinear programming. 3rd Edition, Athena Scientific.
3. Bertsimas, D., & Tsitsiklis, J. N. (1997). Introduction to linear optimization. Belmont, MA: Athena Scientific.
4. Dantzig, G. (2016). Linear programming and extensions. Princeton university press.
5. Luenberger, D. G., & Ye, Y. (1984). Linear and nonlinear programming (Vol. 2). Reading, MA: Addison-wesley.
6. Papadimitriou, C. H., & Steiglitz, K. (1998). Combinatorial optimization: algorithms and complexity. Courier Corporation.
7. Rao, S. S. (2019). Engineering optimization: theory and practice. John Wiley & Sons.
8. Schrijver, A. (2003). Combinatorial optimization: polyhedra and efficiency (Vol. 24, p. 2). Berlin: Springer.
9. Sierksma, G., & Zwols, Y. (2015). Linear and integer optimization: theory and practice. CRC Press.
10. Wolsey, L. A. (2020). Integer programming. John Wiley & Sons
11. Wolsey, L. A., & Nemhauser, G. L. (1999). Integer and combinatorial optimization. John Wiley & Sons.



(a) Bertsimas and Tsitsiklis, (1997)



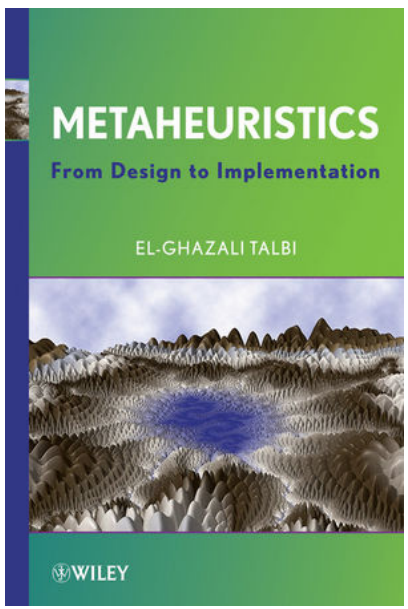
(b) Bertsekas, (2016)



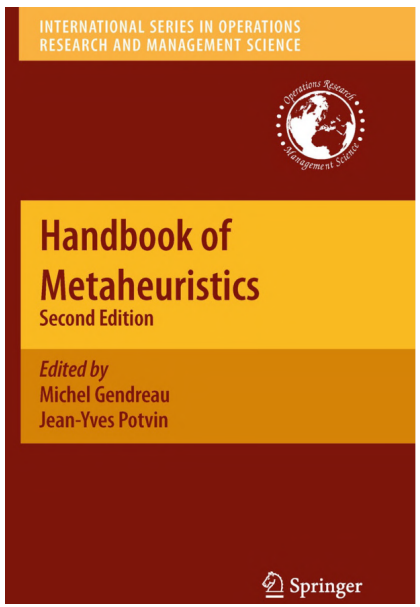
(c) Wolsey, (2020)

Textbook II

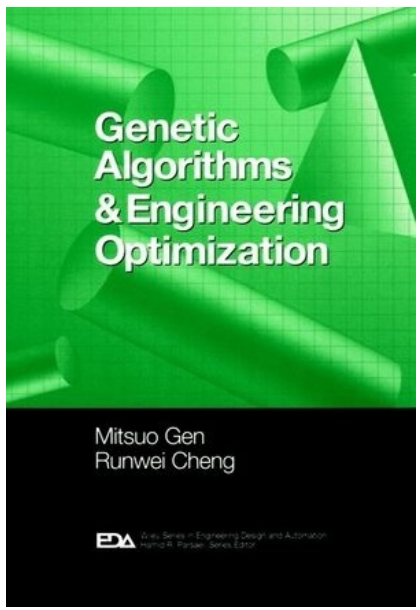
1. Haupt, R. L., & Haupt, S. E. (2004). Practical genetic algorithms. John Wiley & Sons.
2. Gendreau, M., & Potvin, J. Y. (Eds.). (2010). Handbook of metaheuristics. New York: Springer.
3. Gen, M., & Cheng, R. (1999). Genetic algorithms and engineering optimization. John Wiley & Sons.
4. Moustakas, C. (1990). Heuristic research: Design, methodology, and applications. Sage Publications.
5. Talbi, E. G. (2009). Metaheuristics: from design to implementation. John Wiley & Sons.
6. Yu, X., & Gen, M. (2010). Introduction to evolutionary algorithms. Springer Science & Business Media.



(a) Talbi, (2009)



(b) Gendreau & Potvin, (2010)



(c) Gen & Cheng, (1999)

Engineering project

What is an engineering project?

- The branch of science and technology concerned with the design, building, and use of engines, machines, and structures.
- Project in the scope of engineering.

An example engineering project

- Yangshan Deep Water Port Project

- Computer Science
- Industry Engineering
- Project Engineering
- Logistics Engineering
- Management Engineering

- Traffic Engineering
- Ocean Engineering
- Bridge Engineering
- Mechanical Engineering

- Physics
- Material Science
- Electric Engineering
- Mechanical Engineering

- Earth Sciences
- Geological Sciences
- Ecology and Environment

70000 Million RMB

Figure 3: Yangshan deep water port project.

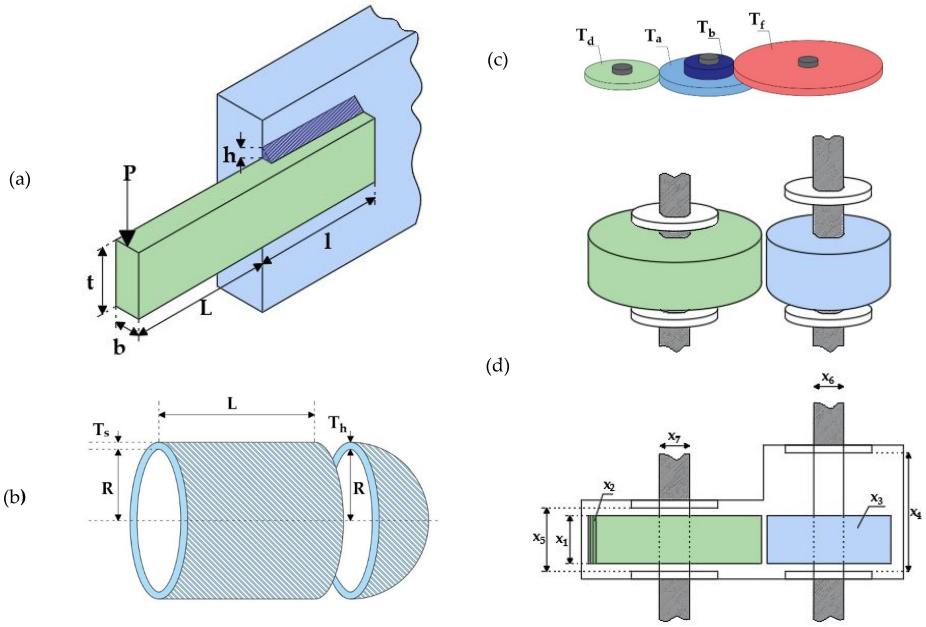
Engineering project

What is an engineering optimization problem?

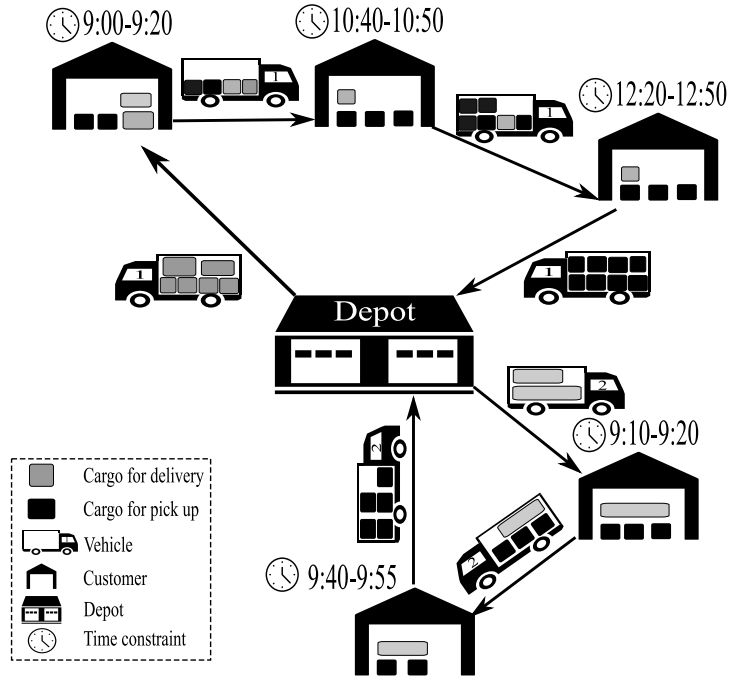
(a) Mechanical engineering related problems

(b) Logistics & transportation related problems.

(c) .....



(a) Uray *et al.* [1]



(b) Shi *et al.* [2]

Figure 4: Engineering optimization problems

Decision problem

Definition of decision problems

- A problem with a "yes" or "no" answer

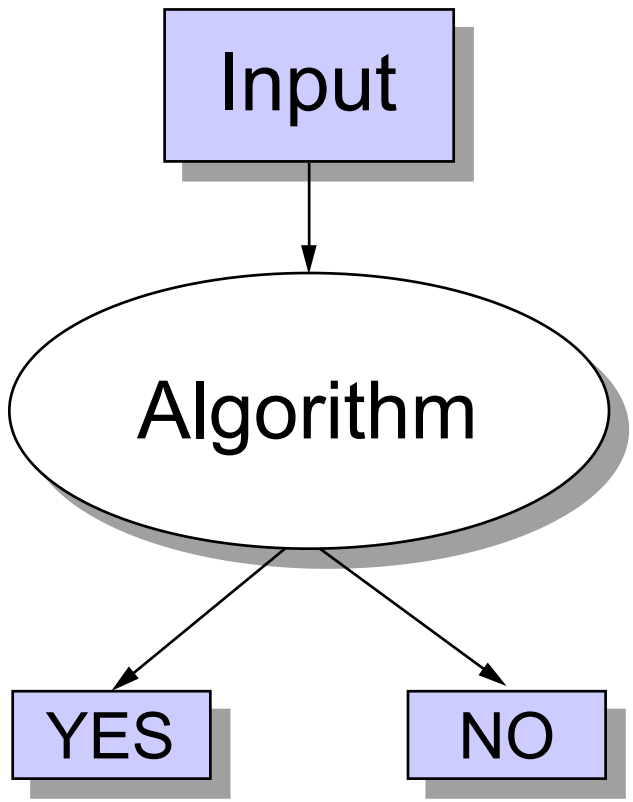


Figure 5: A decision problem has only two possible outputs (yes or no) on any input.[1]

Decision problem

Can all decision problems be solved ?

- ☒ A undecidable problem(不可判定问题)
- ☑ A decidable problem (可判定问题)

An example of undecidable problem

- Tilings of the plane (铺地板问题) Robinson [3]?

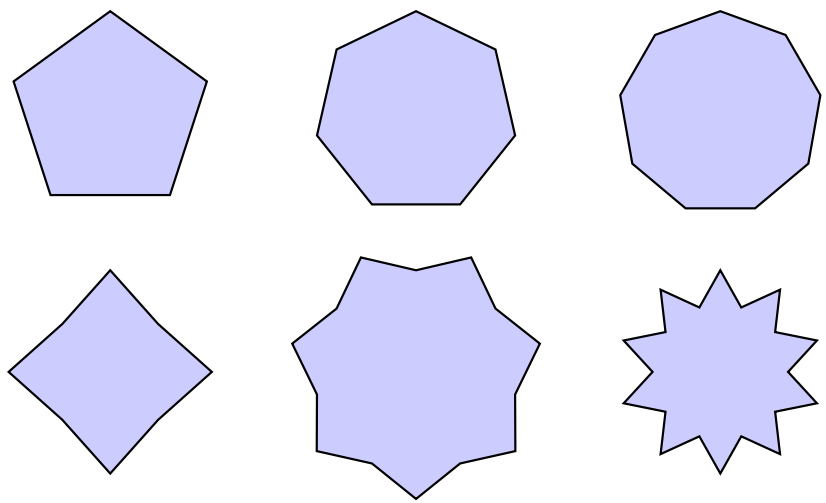


Figure 6: A tilings of the plane problem with 6 types of tilings.

Decision problem

An example of decidable problem

- The Sudoku game shown in figure (b) is solved or not?

	2		5		1		9	
8			2		3			6
	3			6			7	
		1				6		
5	4						1	9
		2				7		
	9			3			8	
2			8		4			7
	1		9		7		6	

(a)Initialized Sudoku

4	2	*6*	5	*7*	1	*3*	9	*8*
8	*5*	*7*	2	*9*	3	*1*	*4*	6
1	3	*9*	*4*	6	*8*	*2*	7	*5*
9	*7*	1	*3*	*8*	*5*	6	*2*	*4*
5	4	*3*	*7*	*2*	*6*	*8*	1	9
6	*8*	2	*1*	*4*	*9*	7	*5*	*3*
7	9	*4*	*6*	3	*2*	*5*	8	*1*
2	*6*	*5*	8	*1*	4	*9*	*3*	7
3	1	*8*	9	*5*	7	*4*	6	*2*

(b) Solved or not?

Decision problems

A decision problem is difficult to solve or not?

- P problem.
- NP problem.
- NP-hard problem.
- NP-complete problem.

Table 1: Verifiable and solvable in polynomial time of P, NP, NP-hard and NP-complete problem.

Problem type	Verifiable in p time	Solvable in p time
P	Yes	Yes
NP	Yes	Yes or No
NP-complete	Yes	Unknown
NP-hard	Yes or No	Unknown

Optimization problems

Definition of optimization problems

- A problem of finding the best solution from all feasible solutions

$$
\begin{aligned}
\underset{x}{\text{minimize}} \quad & f(x) \\
\text{subject to} \quad & g_i(x) \leq 0, \quad i = 1, \ldots, m \\
& h_j(x) = 0, \quad j = 1, \ldots, p \\
& x \in \mathbb{R}^n
\end{aligned}
\tag{1}
$$

Optimization problems

Local optimal and global optimal solution

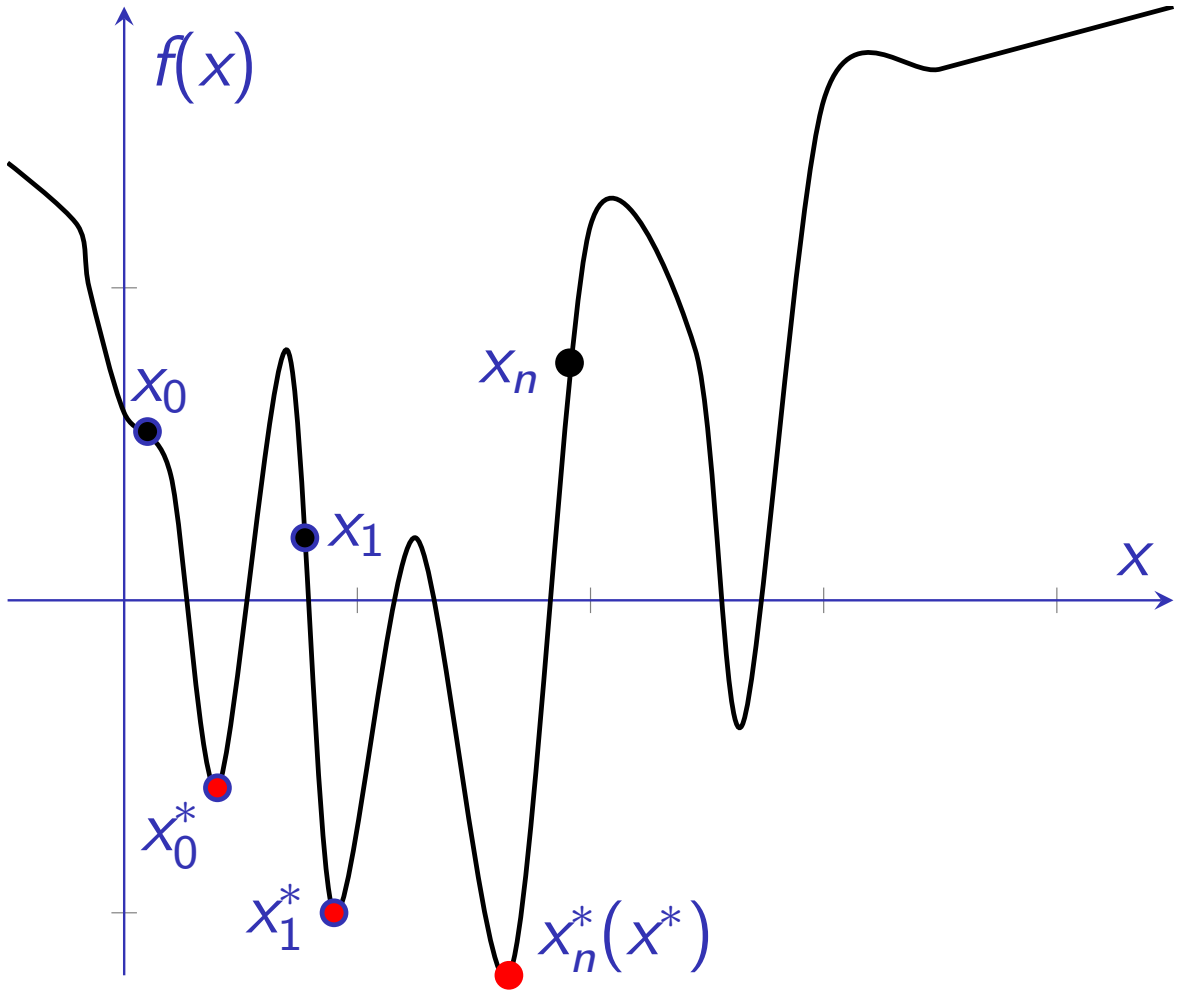


Figure 7: An example of local optimal and global optimal solution.

Decision and optimization problems

An example of decision problems

- How to solve the Sudoku game[2] shown in figure (a)?

	2		5		1		9	
8			2		3			6
	3			6			7	
		1				6		
5	4						1	9
		2				7		
	9			3			8	
2			8		4			7
	1		9		7		6	

(a)Unsolved Sudoku

4	2	*6*	5	*7*	1	*3*	9	*8*
8	*5*	*7*	2	*9*	3	*1*	*4*	6
1	3	*9*	*4*	6	*8*	*2*	7	*5*
9	*7*	1	*3*	*8*	*5*	6	*2*	*4*
5	4	*3*	*7*	*2*	*6*	*8*	1	9
6	*8*	2	*1*	*4*	*9*	7	*5*	*3*
7	9	*4*	*6*	3	*2*	*5*	8	*1*
2	*6*	*5*	8	*1*	4	*9*	*3*	7
3	1	*8*	9	*5*	7	*4*	6	*2*

(b)Solved Sudoku

[2] This example is provided by the https://texample.net/tikz/examples/nontech/games/

Decision and optimization problems

Which is more easier to be solved?

The decision version of a problem is easier than (or the same as) the optimization version.

Optimization problem $\rightarrow$ Decision problem?

Given a solution of an optimization, we can verify that a solution is better than the current best solution or not (decision problem).

Definition of engineering optimization

Engineering optimization is the subject which uses optimization techniques to achieve design goals in engineering [3].

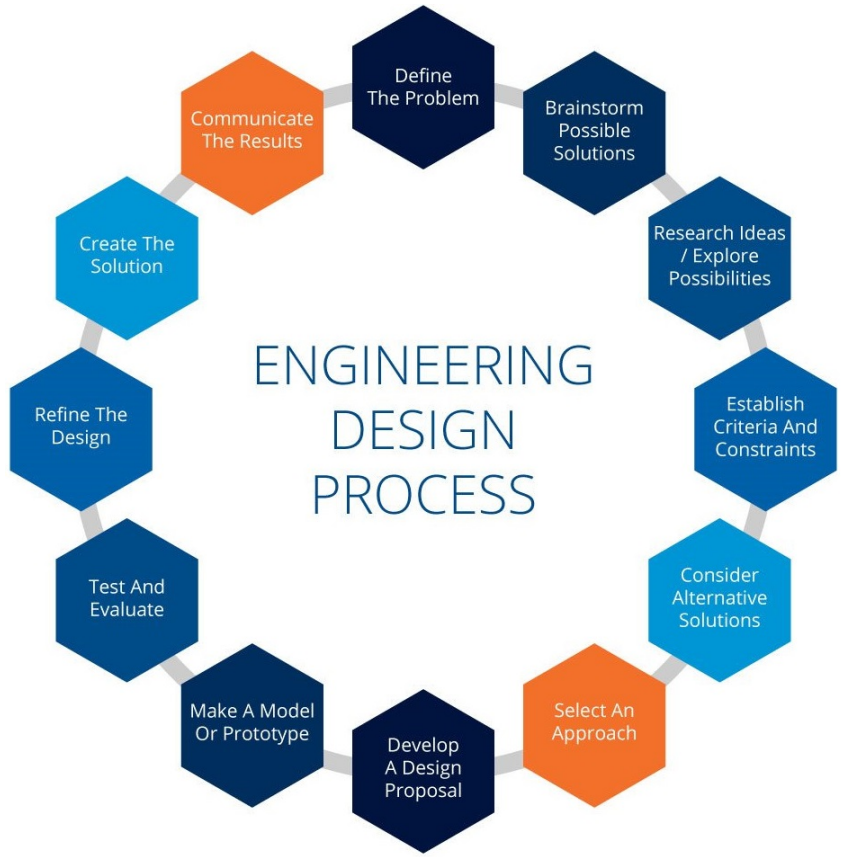


Figure 8: An example of engineering design process [4].

[3] From WIKI: https://en.wikipedia.org/wiki/Engineering_optimization

[4] https://www.twi-global.com/technical-knowledge/faqs/engineering-design-process

Design optimization

Design optimization uses mathematical formulation to obtain the optimal solutions.

$$\begin{array}{lll} \text{minimize} & f(x) & \\ \text{subject to} & h_i(x) = 0, & i = 1, \ldots, m_1 \\ & g_j(x) \leq 0, & j = 1, \ldots, m_2 \\ \text{and} & x \in X \subseteq R^n & \end{array} \tag{2}$$

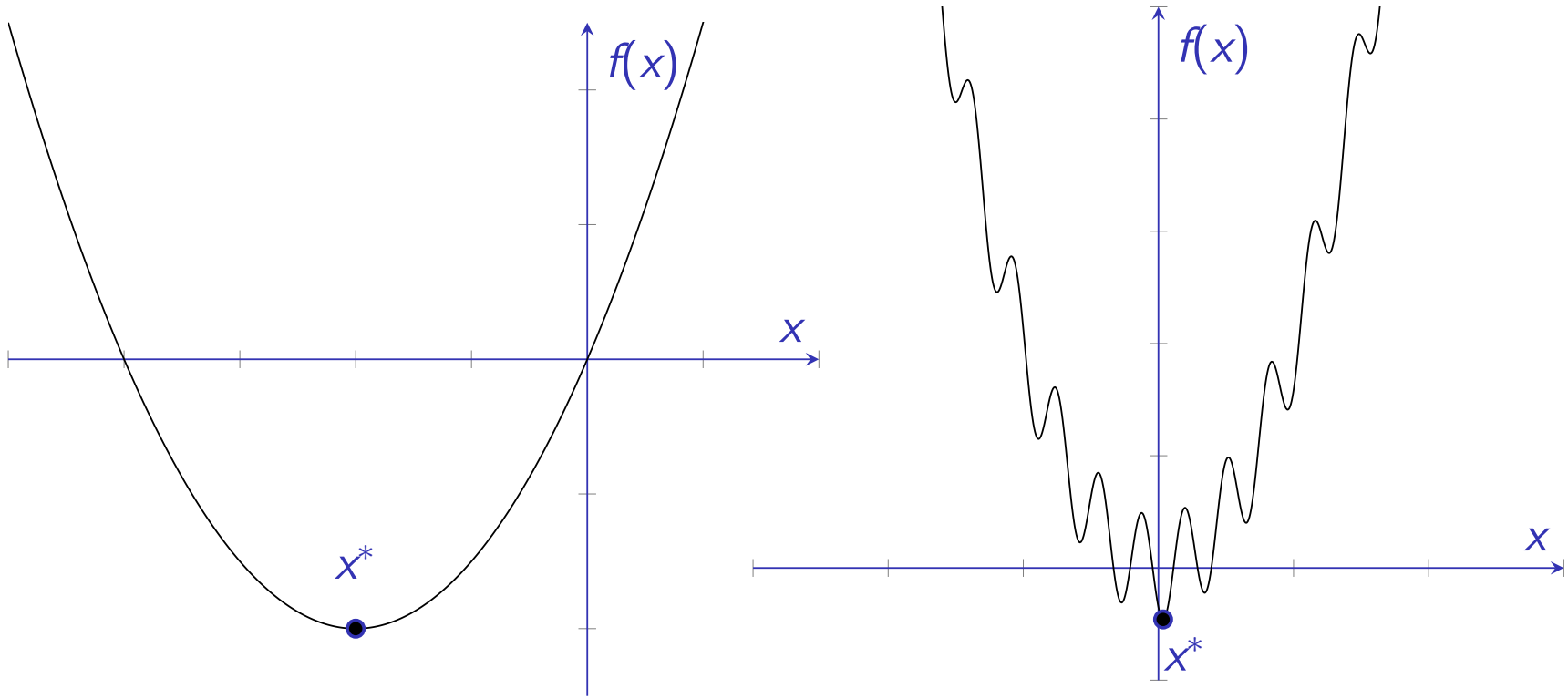
Important considerations

- Decision variable
- Objective function
- Constraint

Design optimization

Optimization problem type

- Convex or Non-Convex



(a) $f(x)=x^2 + 4 * x$ (b) $f(x) = 0.1x^2 - 6sin(x) - 7cos(x)$

Figure 9: Convex and Non-convex continuous smooth functions.

Design optimization

Optimization problem type

- Differentiable vs. non-differentiable

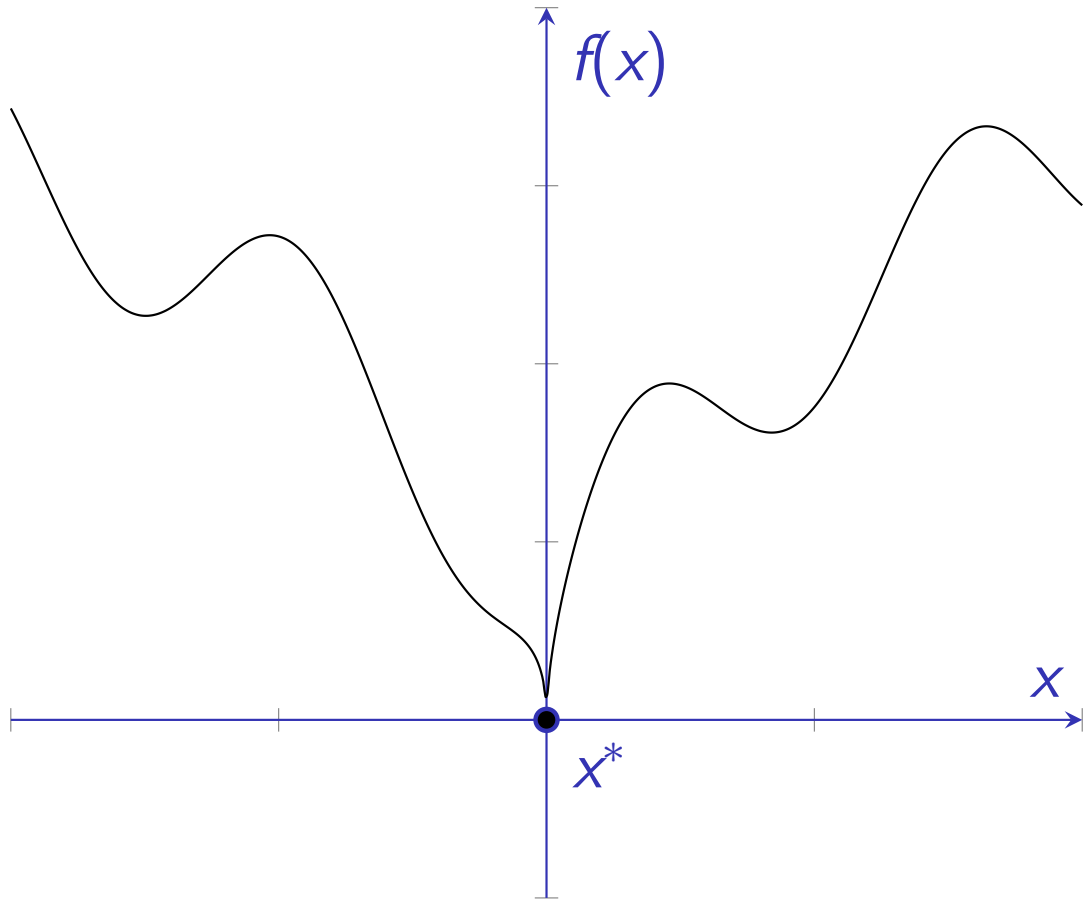


Figure 10: $f(x) = 2|x|^{0.5} + sin(x)$

Design optimization

Optimization problem type

- None, One, or Many Objectives

$$
\begin{aligned}
\text{Max} \quad & Z_1 = -3x_1 - 6x_2 + 3x_3 \\
\text{s.t} \quad & -x_1 + 3x_3 \leq 0 \\
& x_1 - 6x_3 \leq 0 \\
& -x_2 + 3x_3 \leq 0 \\
& x_2 - 6x_3 \leq 0 \\
& x_1, x_2 \geq 0 \\
& x_3 \geq 0 \text{ and integer}
\end{aligned}
$$

$$
\begin{aligned}
\text{Max} \quad & Z_1 = -3x_1 - 6x_2 + 3x_3 \\
\text{Max} \quad & Z_2 = 15x_1 + 4x_2 + x_3 \\
\text{s.t} \quad & -x_1 + 3x_3 \leq 0 \\
& x_1 - 6x_3 \leq 0 \\
& -x_2 + 3x_3 \leq 0 \\
& x_2 - 6x_3 \leq 0 \\
& x_1, x_2 \geq 0 \\
& x_3 \geq 0 \text{ and integer}
\end{aligned}
$$

Design optimization

Optimization problem type

- Unconstrained or Constrained

$$\text{Max} \quad -4x_1^4 - 3x_2^3 + 5x_3^2 + 5$$

$$\begin{aligned} \text{Max} \quad & -4x_1^4 - 3x_2^3 + 5x_3^2 + 5 \\ \text{s.t} \quad & -x_1 + 3x_3 \leq 0 \\ & x_1 - 6x_3 \leq 0 \\ & -x_2 + 3x_3 \leq 0 \\ & x_2 - 6x_3 \leq 0 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

Design optimization

Optimization problem type

- Deterministic or Stochastic

$$\begin{aligned} \text{Max.} \quad & c_1 x_1 + c_2 x_2 \\ \text{s.t} \quad & a_{11} x_1 + a_{12} x_2 \leq b_1 \\ & a_{21} x_1 + a_{22} x_2 \leq b_2 \\ & a_{31} x_1 + a_{32} x_2 \leq b_3 \\ & x_1, x_2 \geq 0 \end{aligned}$$

$$\begin{aligned} \text{Max.} \quad & \tilde{c_1} x_1 + \tilde{c_2} x_2 \\ \text{s.t} \quad & \tilde{a_{11}} x_1 + \tilde{a_{12}} x_2 \leq \tilde{b_1} \\ & \tilde{a_{21}} x_1 + \tilde{a_{22}} x_2 \leq \tilde{b_2} \\ & \tilde{a_{31}} x_1 + \tilde{a_{32}} x_2 \leq \tilde{b_3} \\ & x_1, x_2 \geq 0 \end{aligned}$$

$\tilde{\square}$ denotes a parameter is uncertain and $\square$ represents a parameter.

Design optimization

Optimization problem type

- Multiple level optimization

The first level problem

$$\min_{x \in X} f_1(x, y) = c_1 x + d_1 y \tag{3}$$

$$s.t. A_1 x + B_1 y \leq b_1 \tag{4}$$

$$x \geq 0 \tag{5}$$

The second level problem

$$\min_{y \in Y} f_2(x, y) = c_2 x + d_2 y \tag{6}$$

$$s.t. A_2 x + B_2 y \leq b_2 \tag{7}$$

$$y \geq 0 \tag{8}$$

Design optimization

Optimization techniques

- Exact algorithm
  - Iterative algorithm
  - Enumerative algorithm
- Approximate algorithm
  - Ad Hoc Heuristics
    - Local search
    - Constructive heuristics
  - Heuristics
    - Simulated annealing
    - Hill climbing
    - .....
    - Genetic algorithms
    - Particle swarm optimization
    - .....

A case study: container relocation problem

How to discover an engineering optimization problem?

- Container relocation problem [4].

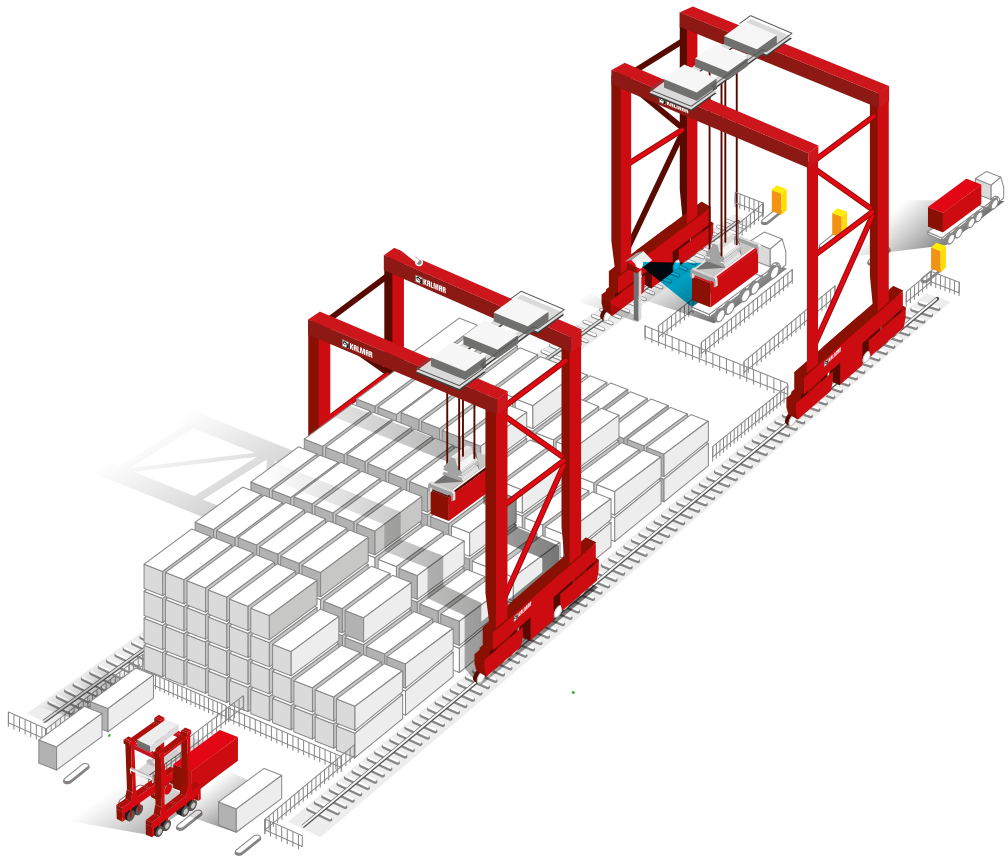


Figure 12: A block and two yard cranes.

A case study: container relocation problem

How to discover an engineering optimization problem?

- Container relocation problem [4].

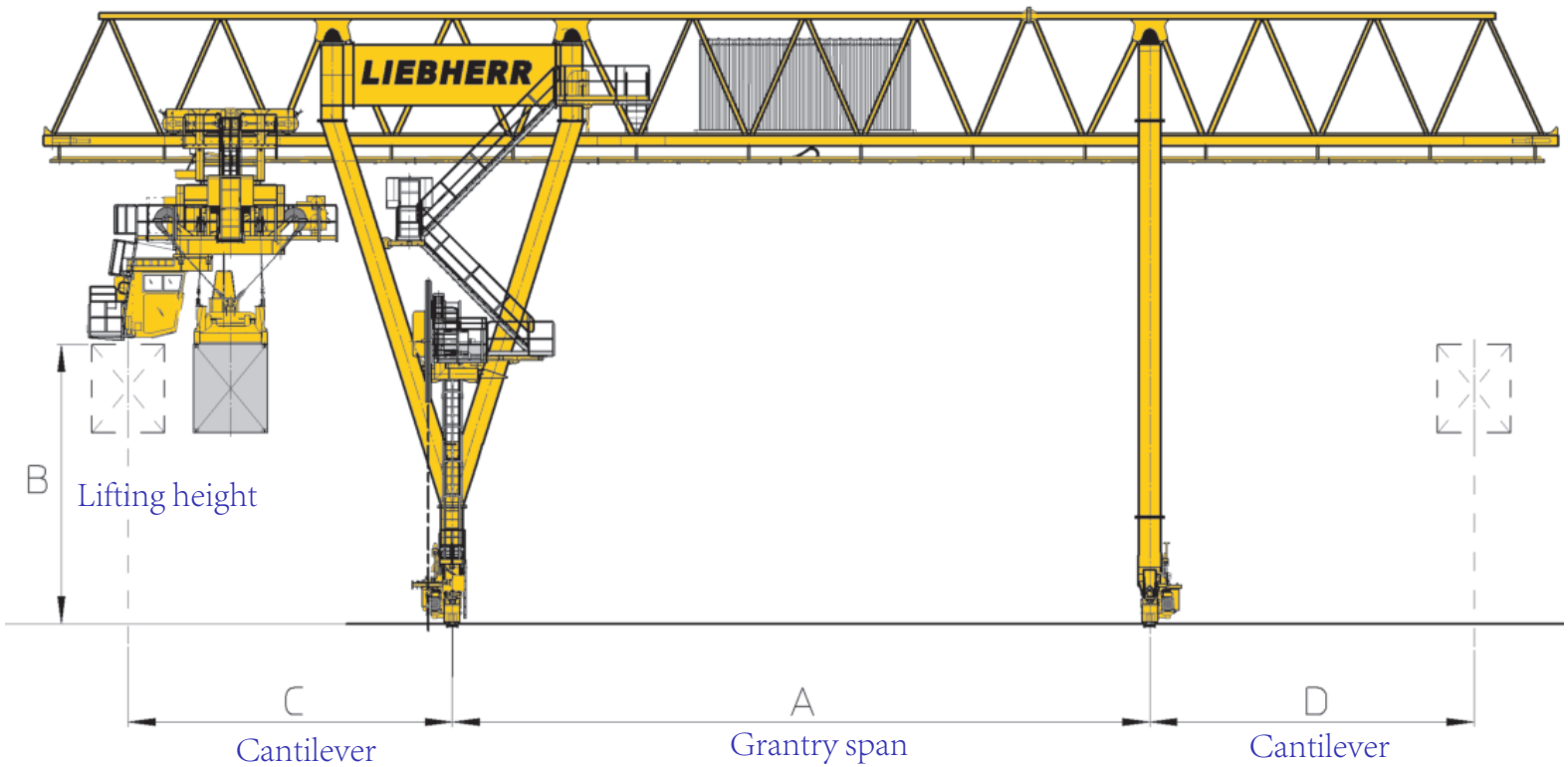


Figure 13: Technical description Liebherr rail mounted gantry crane [5].

[5] Figure source: https://psndealer.com/dealersite/images/newvehicles/2014/nv403816_1.pdf

A case study: container relocation problem

How to define container relocation problem?

- Description of container layout.

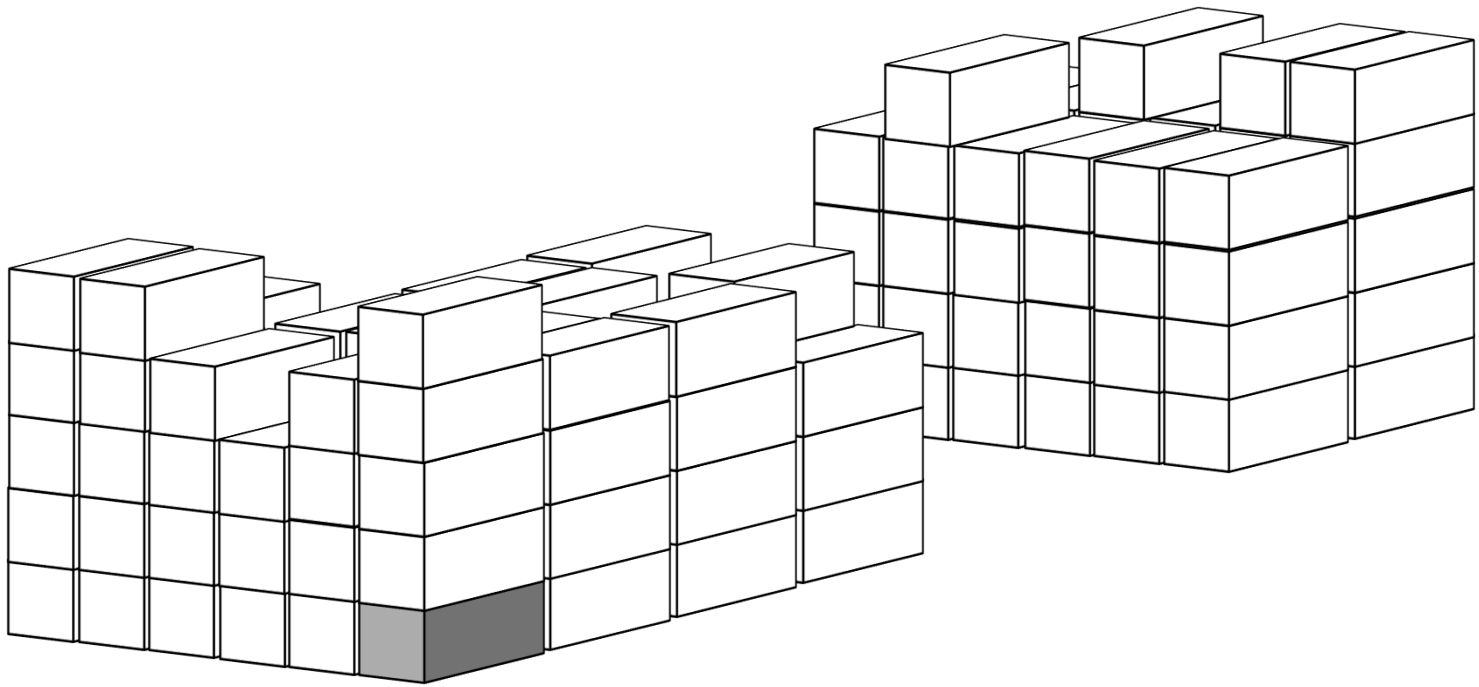


Figure 14: A container layout.

A case study: container relocation problem

How to define container relocation problem?

- Description of container relocation process.

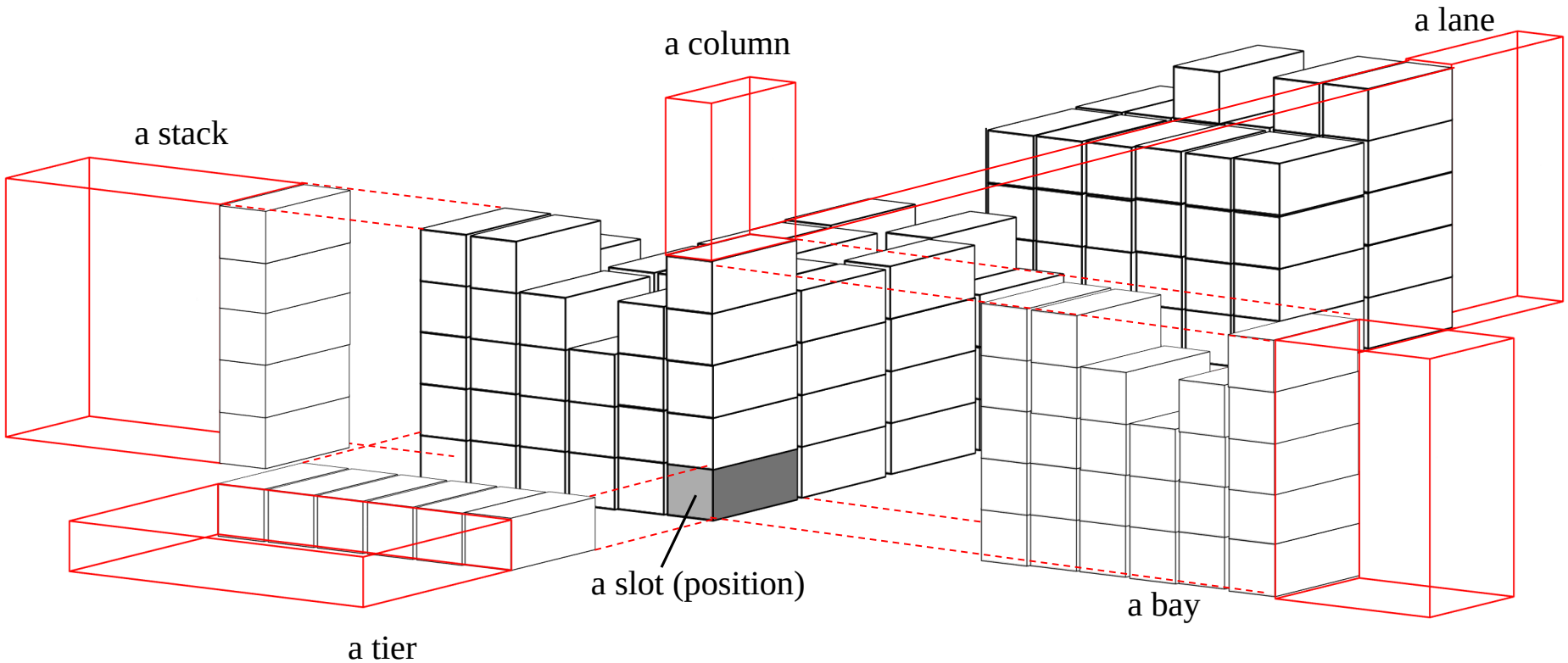


Figure 15: Definition of the container layout.

A case study: container relocation problem

How to define container relocation problem?

- Container relocation process.

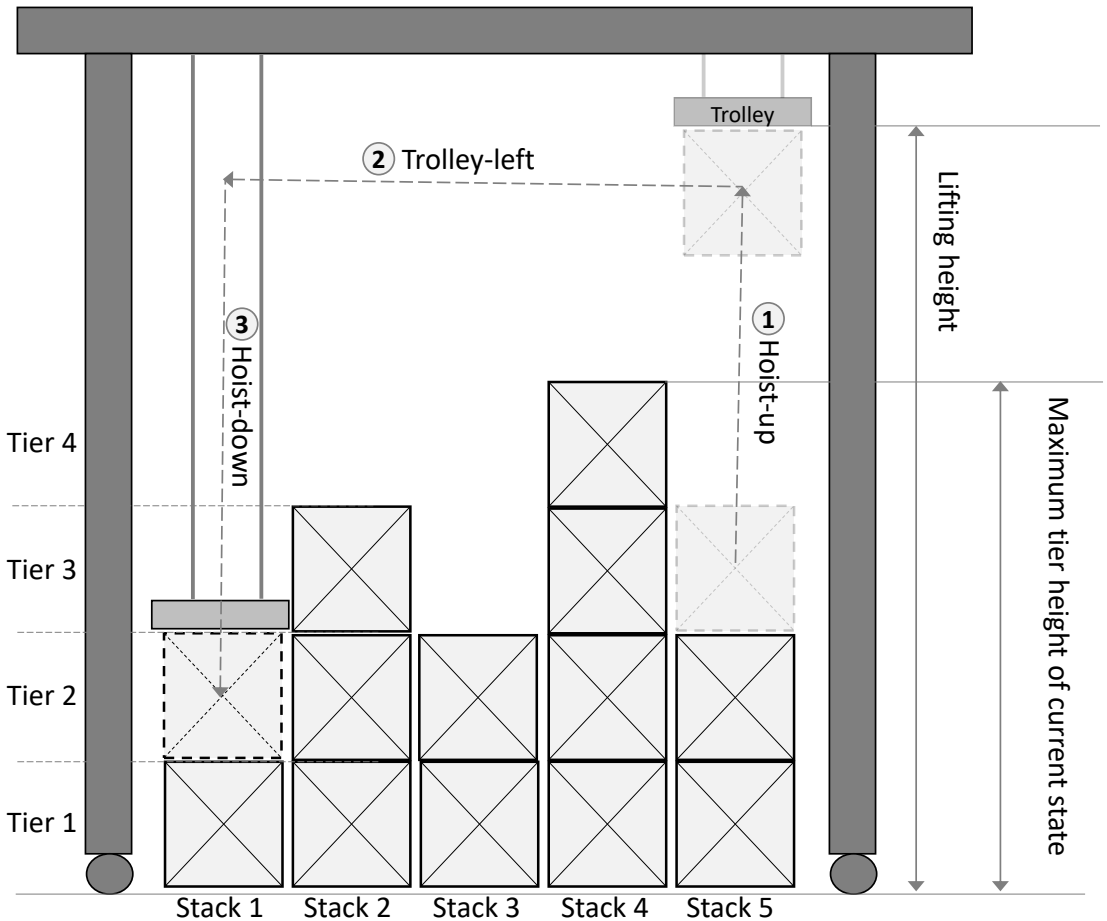


Figure 16: Container relocation process.

A case study: container relocation problem

How to define container relocation problem?

- Matrix representation of container relocation process.

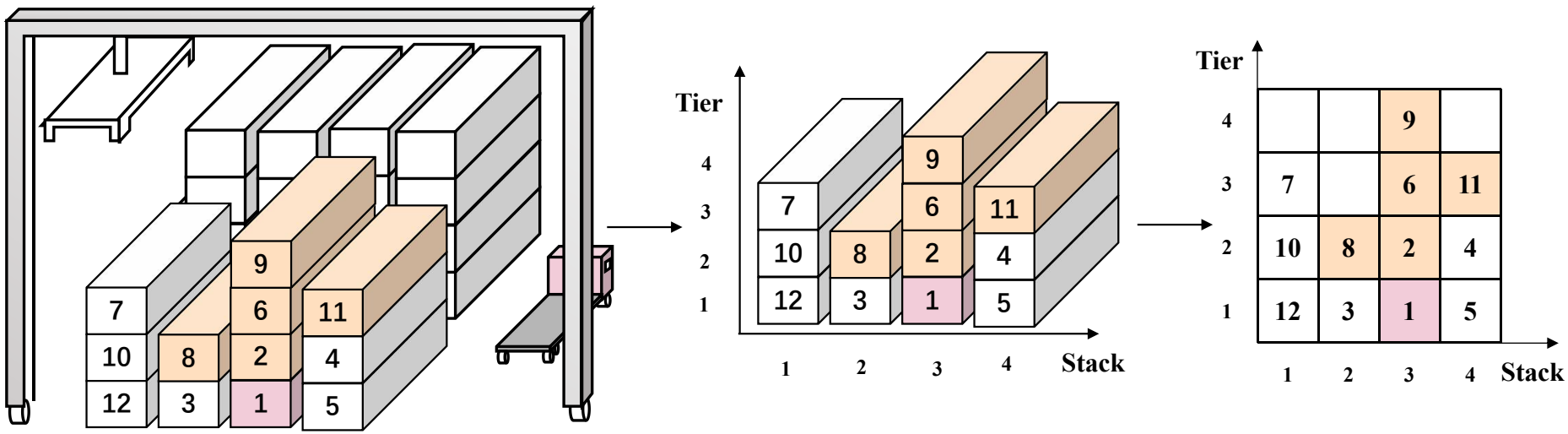


Figure 17: Matrix representation.

A case study: container relocation problem

Goal container relocation problem?

- Minimize the total relocation/location operation of container.

Decision variables of container relocation problem?

- The relocation sequence and relocation positions.

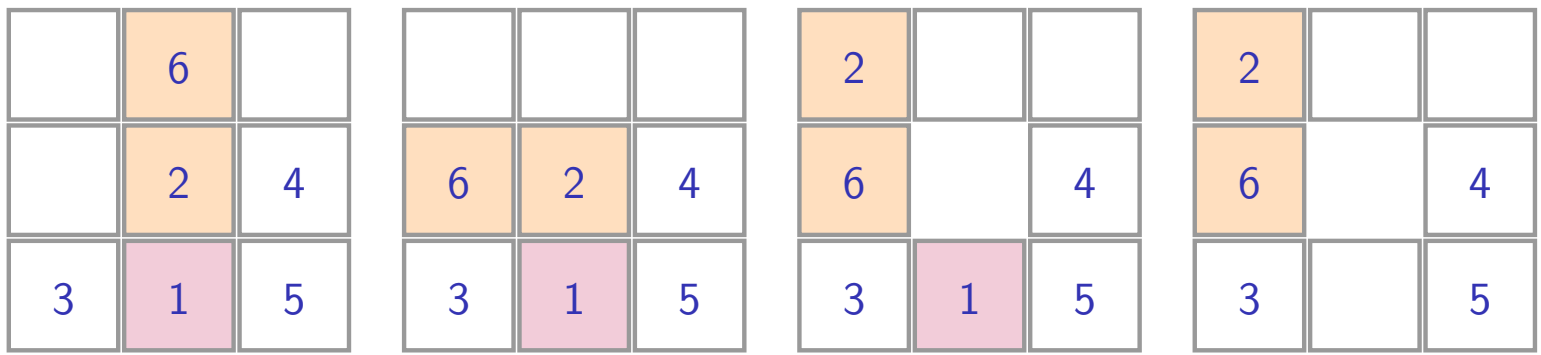


Figure 18: An example of container movement process.

A case study: container relocation problem

Setting the goal of container relocation problem?

(a) Total number of relocation

(b) Total crane movement distance

(c) Operational time of relocation

Coarse-to-Fine

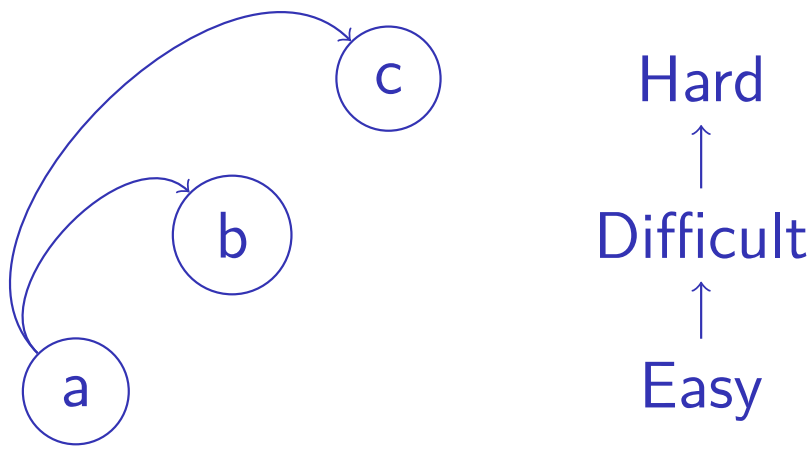


Figure 19: A coarse-to-fine approach for setting the goal of container relocation problem.

A case study: container relocation problem

Setting the goal of container relocation problem?

(a) Total number of relocation

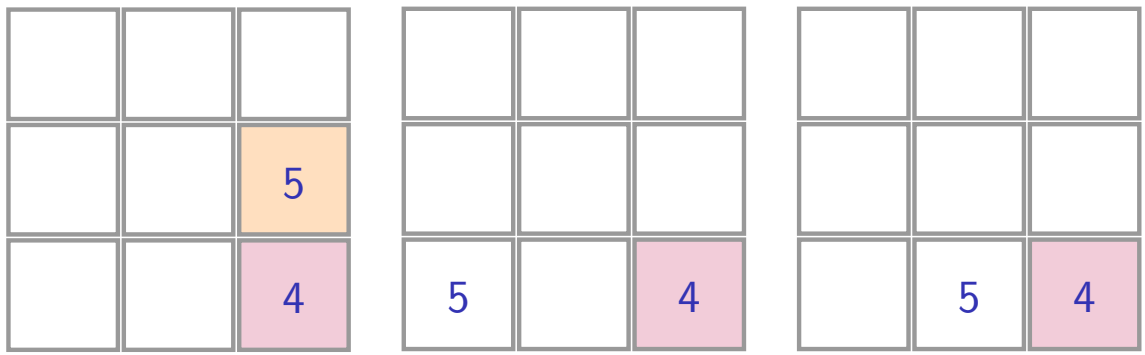


Figure 20: Multiple solutions of container relocation problem.

Pros and Cons

- easy to implement
- multiple solutions

A case study: container relocation problem

Setting the goal of container relocation problem?

(b) Total crane movement distance

- Uniform linear motion

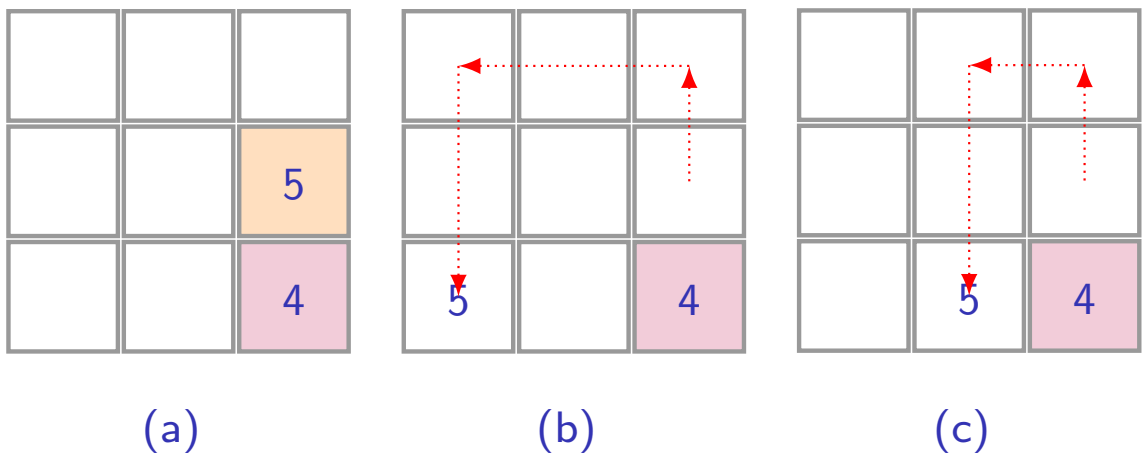


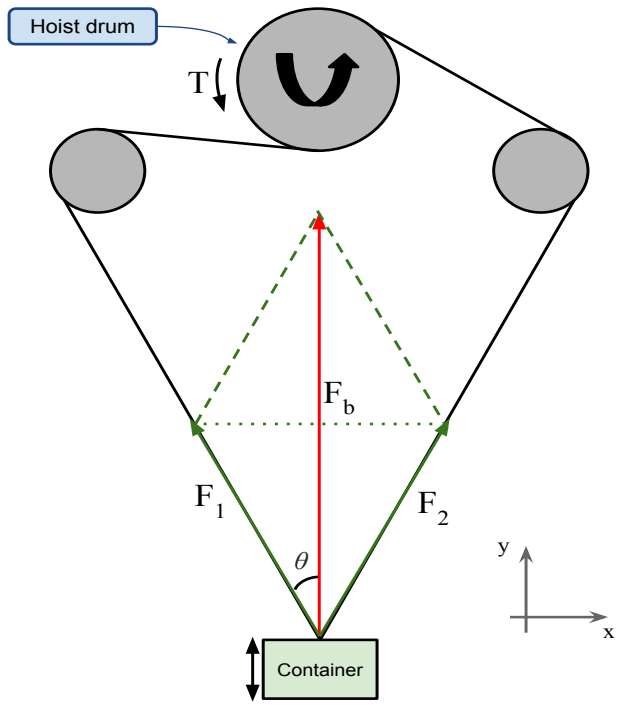
Figure 21: Multiple solutions of container relocation problem.

Pros and Cons?

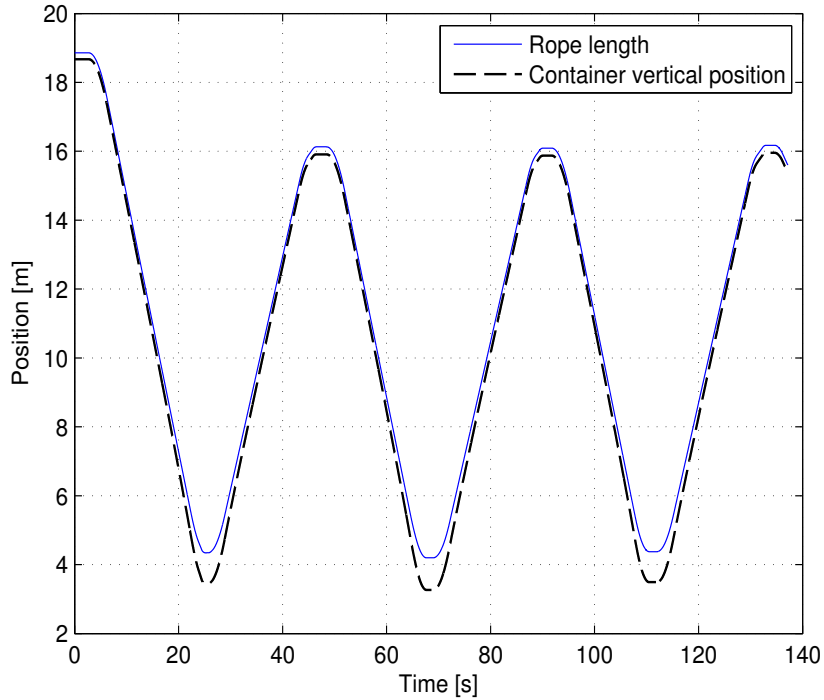
A case study: container relocation problem

Setting the goal of container relocation problem?

(c) Operational time of relocation



(a) Hoist motor movement



(b) Rope position and container vertical position

Figure 22: Crane movement [5].

Pros and Cons?

A case study: container relocation problem

Container relocation problem is difficult to solve or not?

Theorem ([6])

Contianer relocation problem is NP-hard with finite height $H < \infty$ and finite width $W < N$.

N represents the number of containers.

Theorem ([6])

Contianer relocation problem is polynomially solvable when finite width $W \geq N$.

Container relocation problem is NP-hard!

A case study: container relocation problem

How to solve container relocation problem?

- Real time response.
- Exact solution or appropriate solution.

A case study: container relocation problem

Methods to solve container relocation problem

- Mathematical formulation [6–10]
- Exact algorithm
  - Branch and bound [4]
  - Column generation [11]
  - Branch and cut [12]
  - Branch and price [13]
- Meta heuristics.
  - Greedy algorithm [14]
  - Genetic algorithm [15]
  - Ant colony optimization[16]
- Reinforcement learning [17]
- Machine learning [18]

A case study: container relocation problem

Challenges in container relocation problem?

- Uncertainty of container truck?
- Crane trajectory optimization?
- Crane motor optimization?
- .....

Getting started with python

Geometric Interpretation of Linear Programs (GILP):

GILP is a Python package that utilizes Plotly for visualizing the geometry of linear programs and the simplex algorithm[6].

$$\begin{aligned} \text{maximize} \quad & 5x_1 + 3x_2 \\ \text{subject to} \quad & 2x_1 + 1x_2 \leq 20 \\ & 1x_1 + 1x_2 \leq 16 \\ & 1x_1 + 0x_2 \leq 7 \\ & x_1, x_2 \geq 0 \end{aligned} \tag{9}$$

Visualization of a linear model will help you to understand the optimization method.

[6] https://gilp.henryrobbins.com/en/latest/

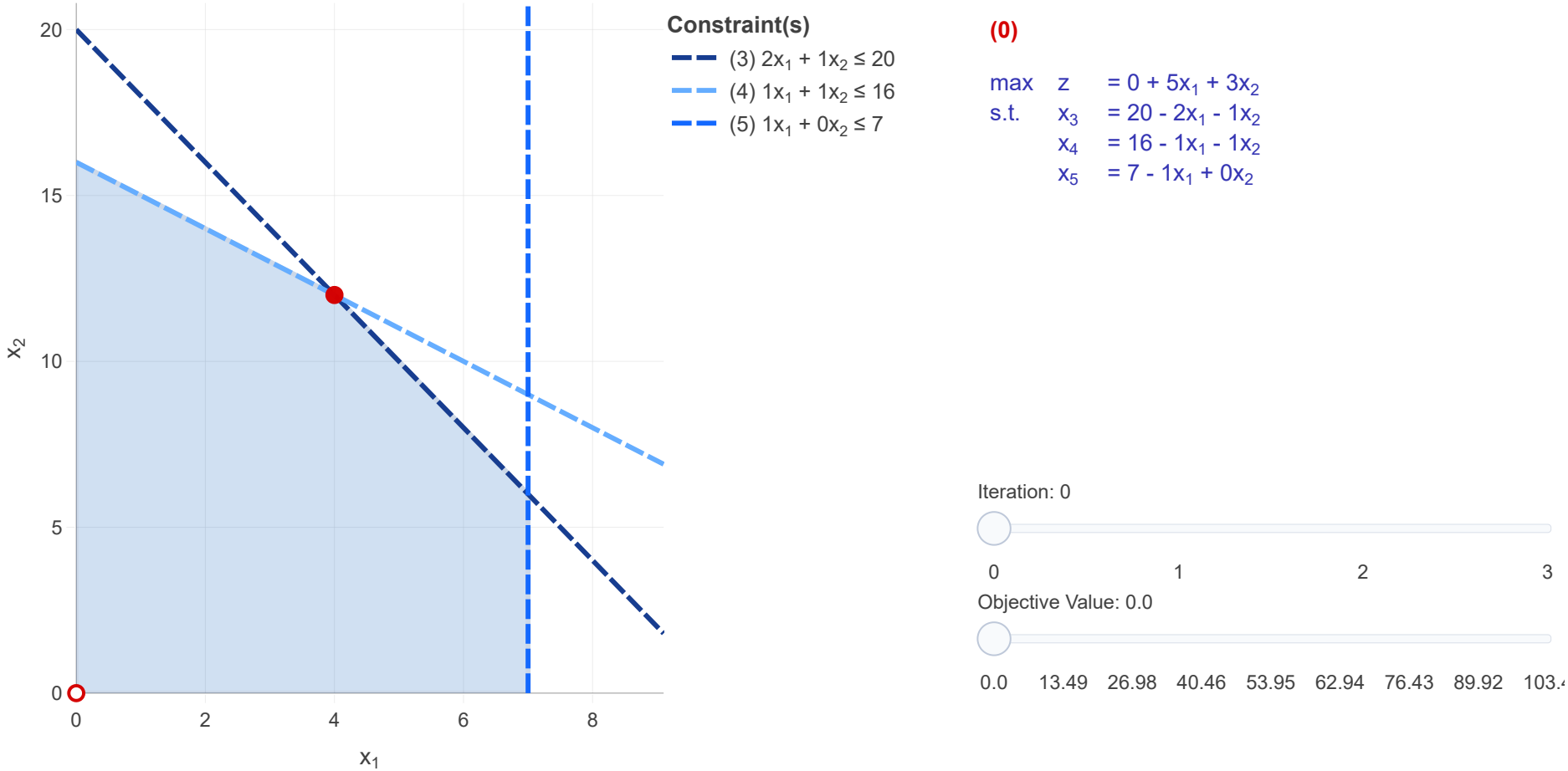


Figure 23: Visualization of the linear model 9

Install python

Python Compiler:

- Offline: https://www.python.org/downloads
- Online: https://www.online-python.com/online_python_compiler

Code editors:

- Visual Studio Code: https://code.visualstudio.com
- PyCharm: https://www.jetbrains.com/pycharm
- Notepad++: https://notepad-plus-plus.org/downloads
- .....

Install GILP

pip install gilp

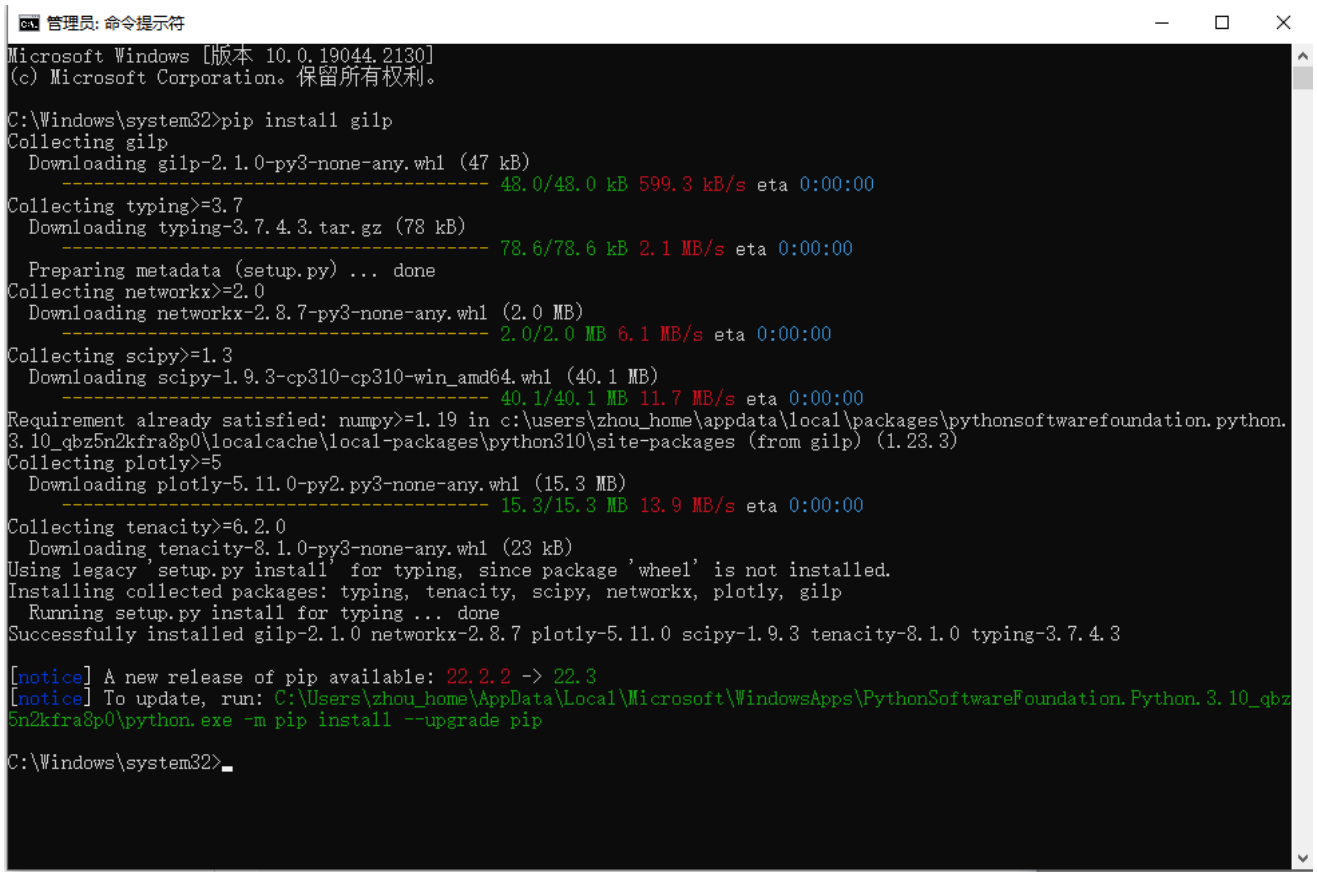


Figure 24: Screen shot of install GILP.

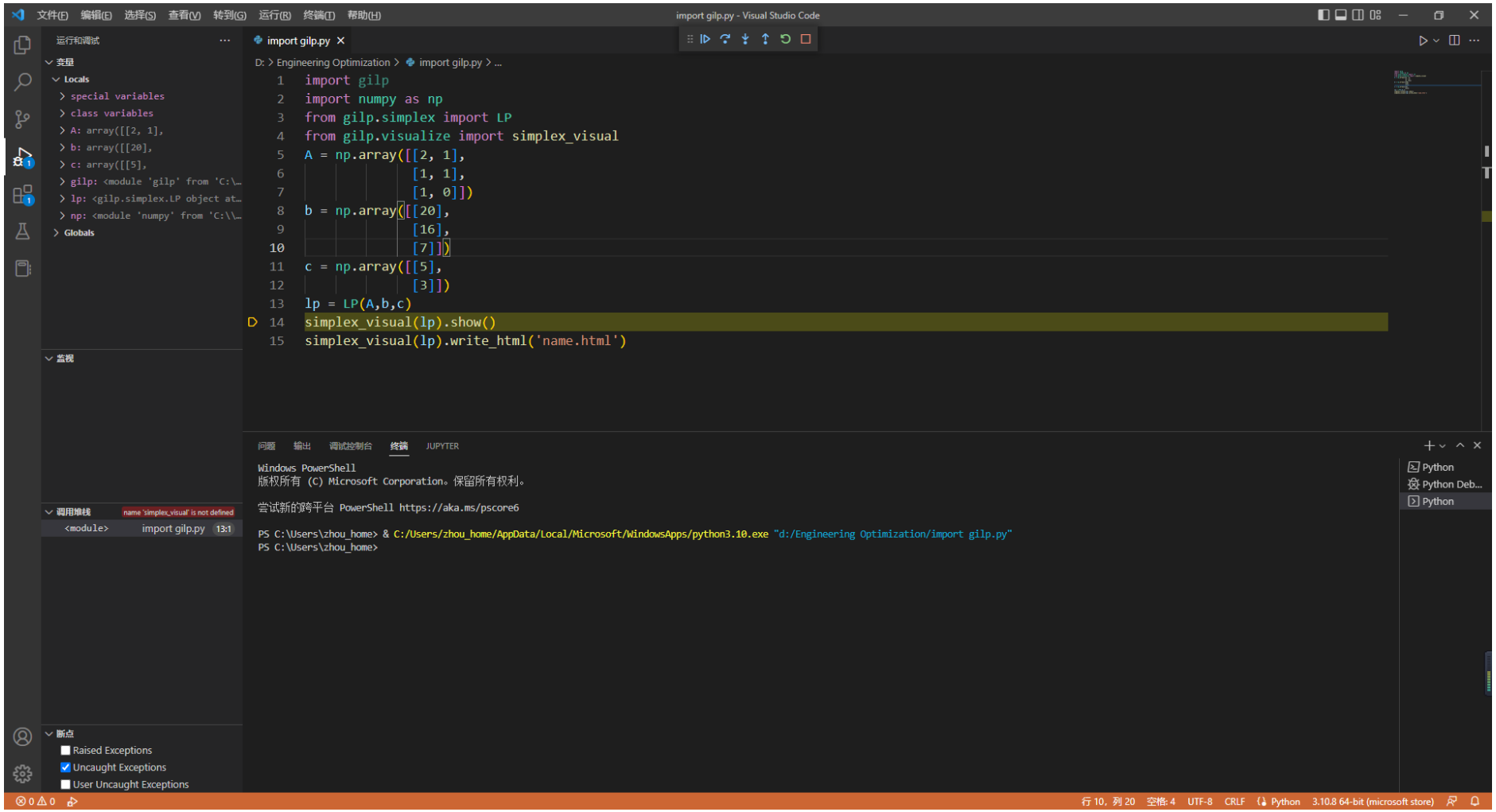


Figure 25: Screen shot of VS Code for testing GILP.

Python-MIP package

Python-MIP is a modelling tool

- Super easy and fast.
- Windows, MacOS and Linux.
- Cut generators, lazy constraints, solution pool, MIP start.

Install

- pip install mip.
- https://www.python-mip.com/.

Python-MIP package

An example of IP model

The 0/1 Knapsack Problem

$$\textit{Maximize} \quad \sum_{i \in I} p_i \cdot x_i \tag{10}$$

$$\textit{Subject to} \quad \sum_{i \in I} w_i \cdot x_i \leq c \tag{11}$$

$$x_i \in \{0, 1\} \forall i \in I \tag{12}$$

Python-MIP package

Modeling 0/1 Knapsack Problem by Python-MIP package [7]

```
from mip import Model, xsum, maximize, BINARY
p = [10, 13, 18, 31, 7, 15]
w = [11, 15, 20, 35, 10, 33]
c, I = 47, range(len(w))
m = Model("knapsack")
x = [m.add_var(var_type=BINARY) for i in I]
m.objective = maximize(xsum(p[i] * x[i] for i in I))
m += xsum(w[i] * x[i] for i in I) <= c
m.optimize()
selected = [i for i in I if x[i].x >= 0.99]
print("selected items: {}".format(selected))
```

[7] https://docs.python-mip.com/en/latest/examples.html

Python tutorials

Books

- Swaroop, C. H. (2013). A Byte of Python. Independent.
- Ramalho, L. (2022). Fluent python. "O'Reilly Media, Inc.".
- Sweigart, A. (2019). Automate the boring stuff with Python: practical programming for total beginners. No Starch Press.

Courses

- https://docs.python.org/3/tutorial/
- http://python.berkeley.edu/resources/
- https://ocw.mit.edu/courses/6-0001-introduction-to-computer-science-and-programmi

Homework 1

Mathematical modeling by python

- Using Python-MIP package https://www.python-mip.com/ to implement the mathematical formulation of the Sudoku game.

Reference paper

- https://langvillea.people.cofc.edu/Sudoku/sudoku2.pdf

References I

Uray, E., Carbas, S., Geem, Z. W. & Kim, S. Parameters optimization of taguchi method integrated hybrid harmony search algorithm for engineering design problems. *Mathematics* **10,** 327 (2022).

Shi, Y., Zhou, Y., Boudouh, T. & Grunder, O. A lexicographic-based two-stage algorithm for vehicle routing problem with simultaneous pickup–delivery and time window. *Engineering Applications of Artificial Intelligence* **95,** 103901 (2020).

Robinson, R. M. Undecidability and nonperiodicity for tilings of the plane. *Inventiones mathematicae* **12,** 177–209 (1971).

Kim, K. H. & Hong, G.-P. A heuristic rule for relocating blocks. *Computers & Operations Research* **33,** 940–954 (2006).

Pietrosanti, S., Holderbaum, W. & Becerra, V. M. *Modelling power flow in a hoist motor of a Rubber Tyred Gantry crane.* in *2015 IEEE Industry Applications Society Annual Meeting* (2015), 1–6.

Caserta, M., Schwarze, S. & Voß, S. A mathematical formulation and complexity considerations for the blocks relocation problem. *European Journal of Operational Research* **219,** 96–104 (2012).

Wan, Y.-w., Liu, J. & Tsai, P.-C. The assignment of storage locations to containers for a container stack. *Naval Research Logistics (NRL)* **56,** 699–713 (2009).

References II

Zehendner, E., Caserta, M., Feillet, D., Schwarze, S. & Voß, S. An improved mathematical formulation for the blocks relocation problem. *European Journal of Operational Research* **245,** 415–422 (2015).

Galle, V., Barnhart, C. & Jaillet, P. A new binary formulation of the restricted container relocation problem based on a binary encoding of configurations. *European Journal of Operational Research* **267,** 467–477 (2018).

Eskandari, H & Azari, E. Notes on mathematical formulation and complexity considerations for blocks relocation problem. **22,** 2722–2728 (2015).

Zehendner, E. & Feillet, D. Column generation for the container relocation problem. (2012).

Bacci, T., Mattia, S. & Ventura, P. A branch-and-cut algorithm for the restricted block relocation problem. *European Journal of Operational Research* **287,** 452–459 (2020).

Zehendner, E. & Feillet, D. A branch and price approach for the container relocation problem. *International Journal of Production Research* **52,** 7159–7176 (2014).

Jin, B., Zhu, W. & Lim, A. Solving the container relocation problem by an improved greedy look-ahead heuristic. *European Journal of Operational Research* **240,** 837–847 (2015).

References III

Hussein, M. I. & Petering, M. E. *Global retrieval heuristic and genetic algorithm in block relocation problem.* in *IIE Annual Conference. Proceedings* (2012), 1.

Jovanovic, R., Tuba, M. & Voß, S. An efficient ant colony optimization algorithm for the blocks relocation problem. *European Journal of Operational Research* **274,** 78–90 (2019).

Jiang, T., Zeng, B., Wang, Y. & Yan, W. *A new heuristic reinforcement learning for container relocation problem.* in *Journal of Physics: Conference Series* **1873** (2021), 012050.

Zhang, C., Guan, H., Yuan, Y., Chen, W. & Wu, T. Machine learning-driven algorithms for the container relocation problem. *Transportation Research Part B: Methodological* **139,** 102–131 (2020).