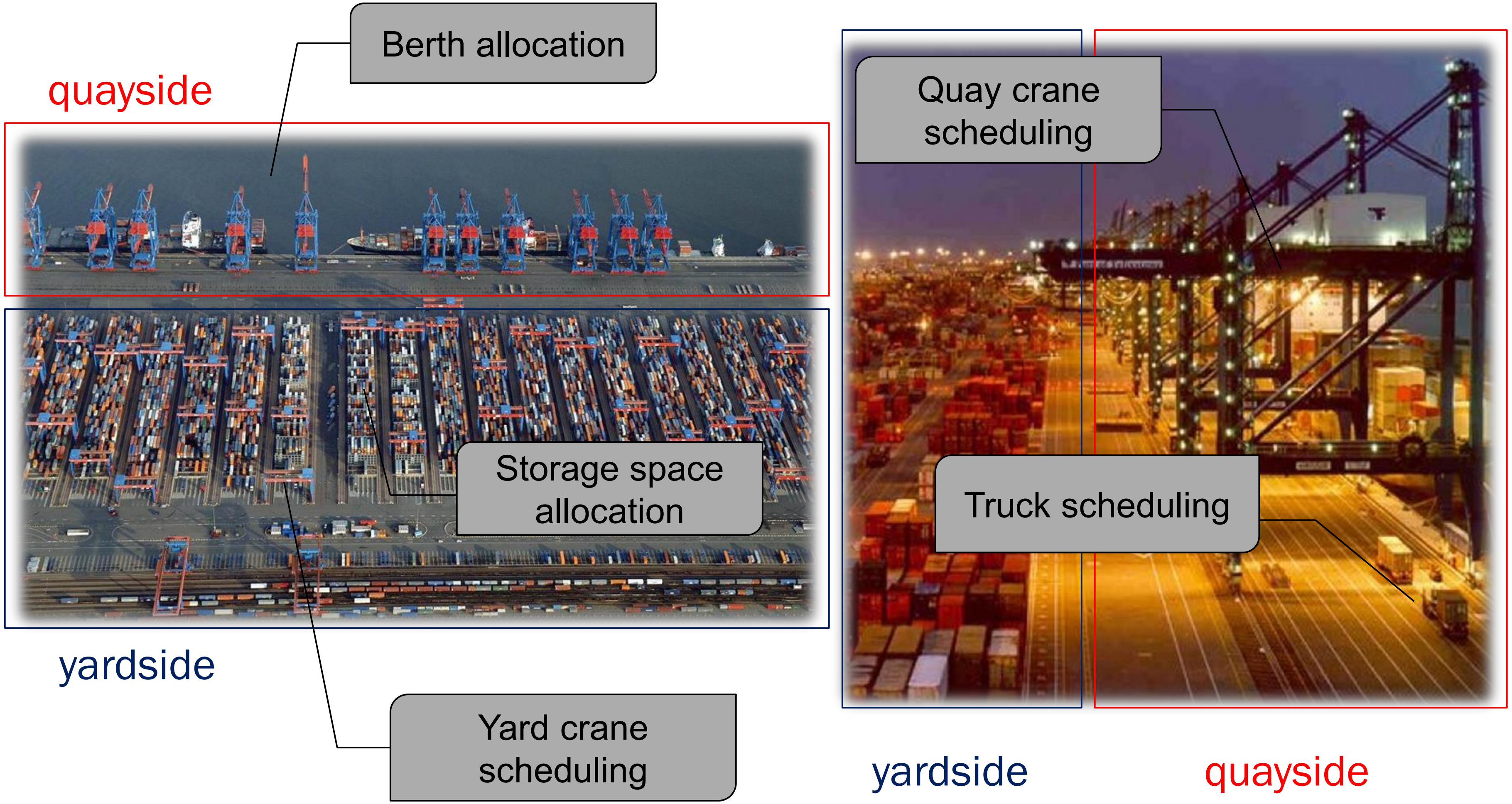
Berth Allocation Problem

- Complex system
- Making decision (Modeling)
  - Allocation
  - Assignment
  - Scheduling
- How to make the wise decisions? (Solving)
  - Deterministic or stochastic?
  - Objective(s)
  - Constraints
- Berth Allocation Problem (BAP)

Berth Allocation Problem



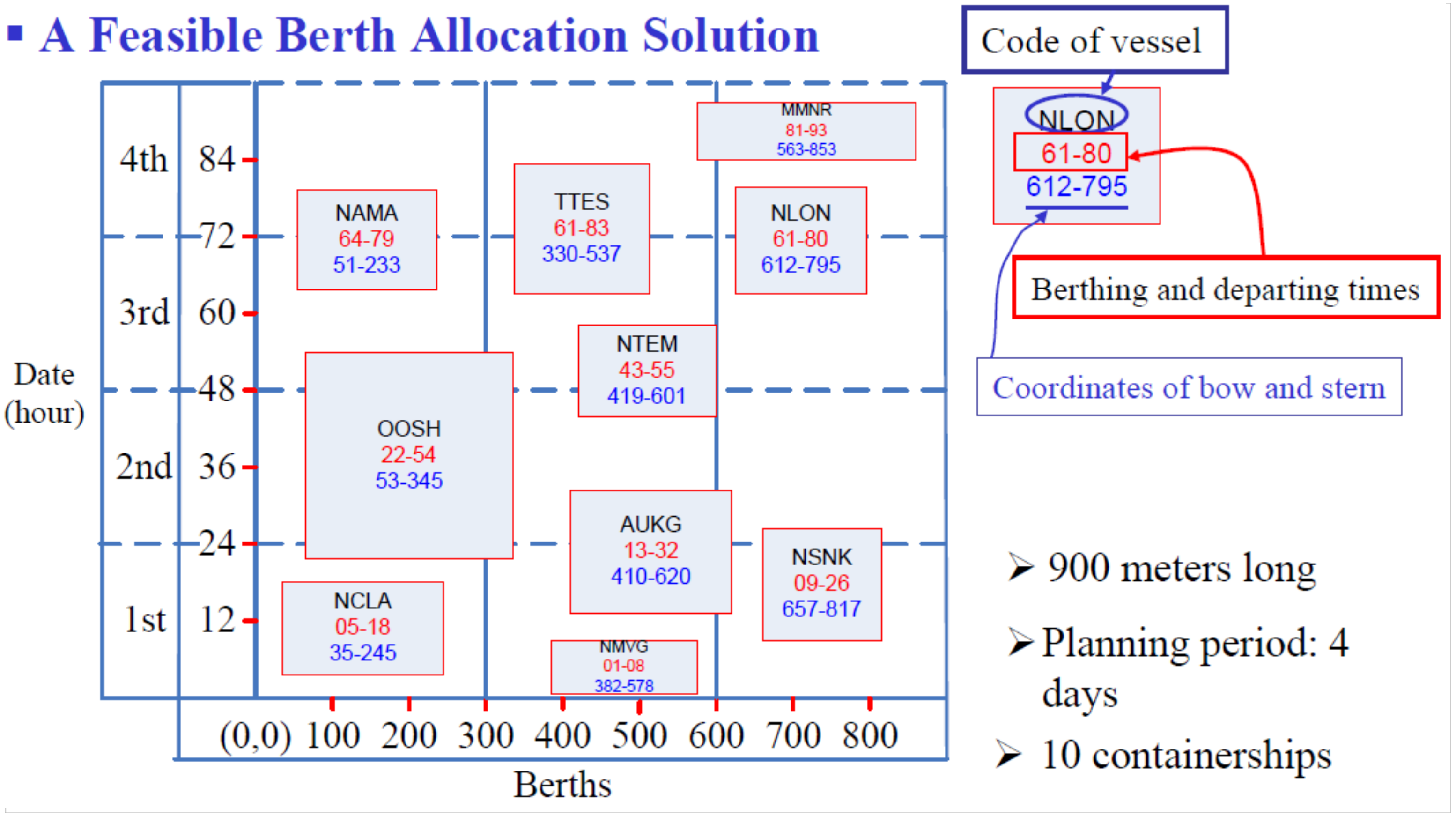
Berth Allocation Problem

- Definition
  - Berth allocation problem aims to determine berthing time and position of each containership that will arrive at a container terminal during a planning period for optimizing an objective.

- Factors being considered
  - Number of available berths
  - Length, arrival time and departure time of each containership
  - Locations and amount of containers for unloading and loading of each containership
- Objectives:
  - Distance: minimize shore to yard distances for all containers
  - Time: minimize departure delay of all containerships

Berth Allocation Problem

- Decision: berthing position; berthing time



Berth Allocation Problem

- Input Parameters

▪ Input Parameters

L: Length of total berth sections (straight line)

l: Number of vessels

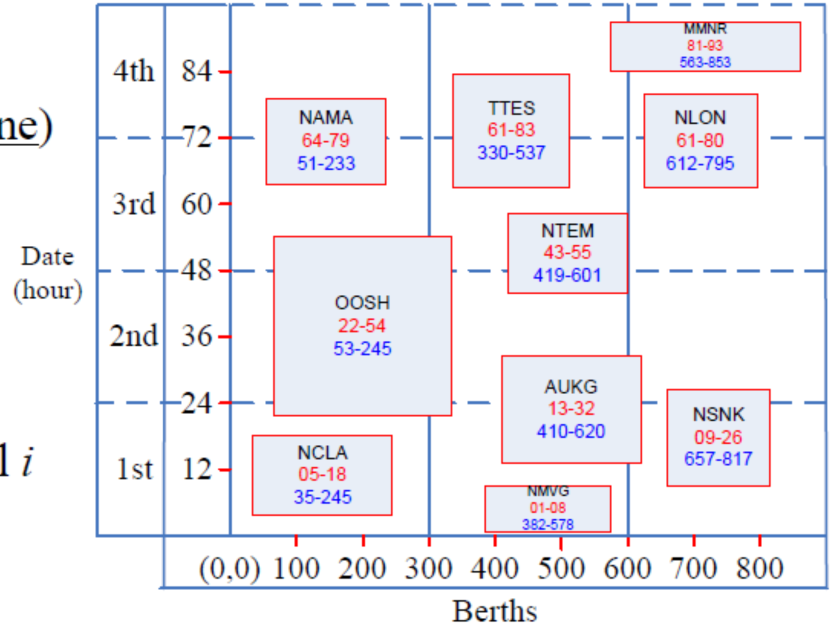
p_i: Best berth location of vessel i

a_i: Expected arrival time of vessel i

b_i: Ship operational time required for vessel i

d_i: Requested departure time of vessel i

l_i: length of vessel i



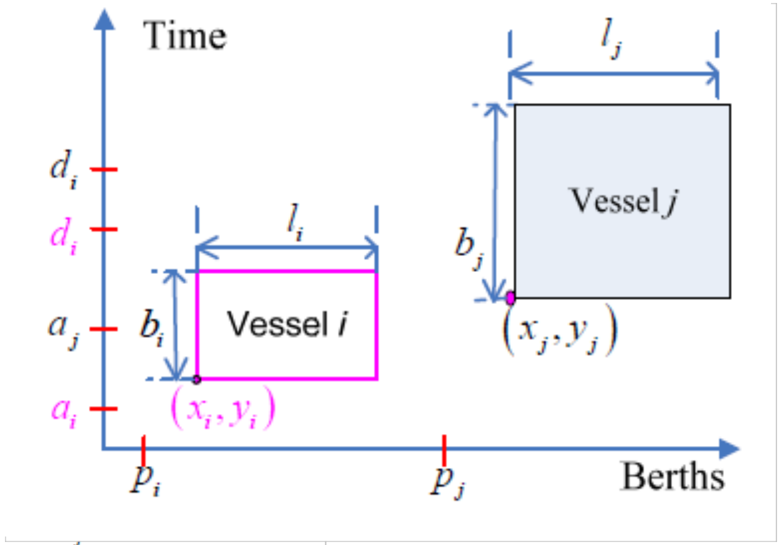
c_{1i}: Additional travel (per one grid-width) cost for delivering containers to vessel i resulting from non-optimal berthing locations.

c_{2i}: Penalty cost (per one grid-length of time) of vessel i resulting from a delayed departure past the requested due time.

Berth Allocation Problem

- Decision Variables

  - x_i: Berth location of vessel i
  - y_i: Berthing time of vessel i



- They aim to express the position relation between any two vessels in the berth-time space:

$$z_{ij}^x = \begin{cases} 1, & \text{if vessel } i \text{ located to the left-hand side of vessel } j \\ 0, & \text{otherwise} \end{cases} \iff x_i + l_i < x_j$$

$$z_{ij}^y = \begin{cases} 1, & \text{if vessel } i \text{ located below vessel } j \\ 0, & \text{otherwise} \end{cases} \iff y_i + b_i < y_j$$

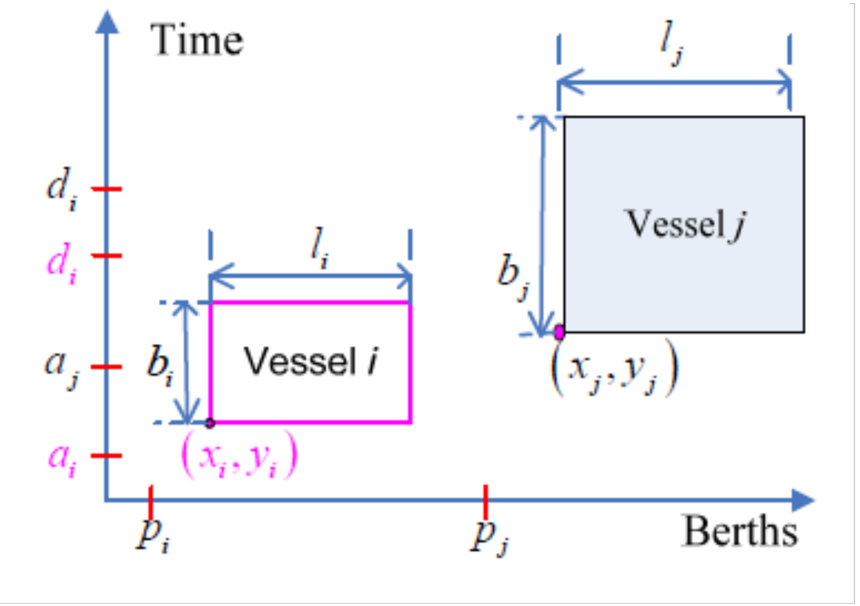
$i=1,2,\ldots,l;\ j=1,2,\ldots,l,$

Berth Allocation Problem

- Objective function

$$\min f(x,y,z)=\sum_{i=1}^{l}\left\{c_{1i}\left|x_i-p_i\right|+c_{2i}\left(y_i+b_i-d_i\right)^+\right\}$$

Note: $[x]^+=\max(0,x)$



- Constraints

$x_i+l_i \leq L,\ i=1,2,\ldots,l$ — The rightmost end of vessel i is restricted by the length of the total berth sections

$$x_i+l_i \leq x_j+M\left(1-z_{ij}^x\right), i=1,\cdots,l;\ j=1,\cdots,l; i\neq j$$

Vessel i is located to the left-hand side of vessel j

$$y_i+b_i \leq y_j+M\left(1-z_{ij}^y\right), i=1,\cdots,l;\ j=1,\cdots,l; i\neq j$$

Departure time of vessel i is not later than the berthing time of vessel j

Berth Allocation Problem

- Constraints

$$z_{ij}^x + z_{ji}^x + z_{ij}^y + z_{ji}^y \geq 1, i = 1, \cdots, l; j = 1, \cdots, l, i < j$$

It excludes the case: $z_{ij}^x + z_{ji}^x + z_{ij}^y + z_{ji}^y = 0$ in which case the rectangles representing schedules for vessels i and j overlap with each other

$$y_i \geq a_i, i = 1, \cdots, l$$

A vessel cannot berth before it arrives

$$x_i \geq 0, i = 1, \cdots, l$$

Non-negativity

$$z_{ij}^x, z_{ij}^y = 0, 1, i = 1, \cdots, l; j = 1, \cdots, l; i \neq j$$

0-1 constraints

Berth Allocation Problem

- Mixed Integer Nonlinear Programming Model

$$\min f(x,y,z) = \sum_{i=1}^{l} \left\{ \boxed{c_{1i} \left| x_i - p_i \right|} + c_{2i} \boxed{(y_i + b_i - d_i)^+} \right\}$$

subject to

$$x_i + l_i \le L, i = 1, 2, \cdots, L$$

$$x_i + l_i \le x_j + M\left(1 - z_{ij}^x\right), i = 1, \cdots, l; j = 1, \cdots, l; i \ne j$$

$$y_i + b_i \le y_j + M\left(1 - z_{ij}^y\right), i = 1, \cdots, l; j = 1, \cdots, l; i \ne j$$

$$z_{ij}^x + z_{ji}^x + z_{ij}^y + z_{ji}^y \ge 1, i = 1, \cdots, l; j = 1, \cdots, l, i < j$$

$$y_i \ge a_i, i = 1, \cdots, l$$

$$x_i \ge 0, i = 1, \cdots, l$$

$$z_{ij}^x, z_{ij}^y = 0, 1, i = 1, \cdots, l; j = 1, \cdots, l; i \ne j$$

Berth Allocation Problem

- Homework
  - Create a test instance by randomly generating the input parameters
  - Develop a mixed integer linear programming model
  - Solve the model by PuLP/CPLEX/Gurobi
  - Present the solution and conduct analysis

Reference

- Optimization with PuLP: https://coin-or.github.io/pulp/index.html

- Main Topics
  - The Optimisation Process
  - Optimisation Concepts
  - Basic Python Coding
  - Installing PuLP at Home
  - Amply
  - README
- Case Studies
  - A Blending Problem
  - A Set Partitioning Problem
  - A Sudoku Problem formulated as an LP
  - A Transportation Problem
  - A Two Stage Production Planning Problem
- User Guides
  - How to configure a solver in PuLP
  - How to warm-start a solver
  - Elastic Constraints
  - How to export models in PuLP
  - How to debug most errors during solving
- PuLP Internal Documentation
  - `pulp.constants`
  - `pulp` : Pulp classes
  - `pulp.apis` Interface to Solvers
- Plugins
  - orloge: OR logs parser
  - pytups: smart dictionaries and tuple lists