Fundamentals of Optimization

Introduction

Pham Quang Dung

dungpq@soict.hust.edu.vn

Department of Computer Science

Content

- Optimization problems
- Optimization problem classification
- Applications
- Topics

Optimization problems

- Maximize or minimize some function relative to some set (range of choices)
- The function represents the quality of the choice, indicating which is the "best"
- Example
  - A shipper need to find the shortest route to deliver packages to customers 1, 2, …, N

Notations

- $x \in R^n$: vector of decision variables $x_{j,}$ $j = 1, 2, \ldots, n$
- $f : R^n \rightarrow R$ is the objective function
- $g_i : R^n \rightarrow R$ is the constraint function defining restriction on x, $i = 1, 2, \ldots, m$

minimize $f(x)$ over $x = (x_1, x_2, \ldots, x_n) \in X \subset R^n$
satisfying a property P:

$$g_i(\mathrm{x}) \leq b_i,\ i = 1, 2, \ldots, s$$
$$g_i(\mathrm{x}) = d_i,\ i = s + 1, 2, \ldots, m$$

Examples

$$\min f(x) = 3x_1 - 5x_2 + 10x_3$$
$$x_1 + x_2 + x_3 \leq 10$$
$$2x_1 + 4x_2 - 5x_3 = 9 \qquad \text{(Linear Program)}$$
$$x_1, x_2 \in R+, x_3 \in Z$$

$$\min f(x) = 4x_1^2 + 3x_2^2 - 7x_1 x_3$$
$$x_1 + x_2^3 + 4x_3 \leq 10$$
$$2x_1^2 + 4x_2 - 5x_3 = 7 \qquad \text{(Nonlinear Program)}$$
$$x_1, x_2 \in R+, x_3 \in Z$$

Solving optimization problems

- General optimization problems
  - Very difficult to solve
- Some special cases
  - Linear programming
  - Least square problem
  - Some shortest path problems on networks
  - Etc.

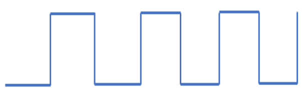
Classification

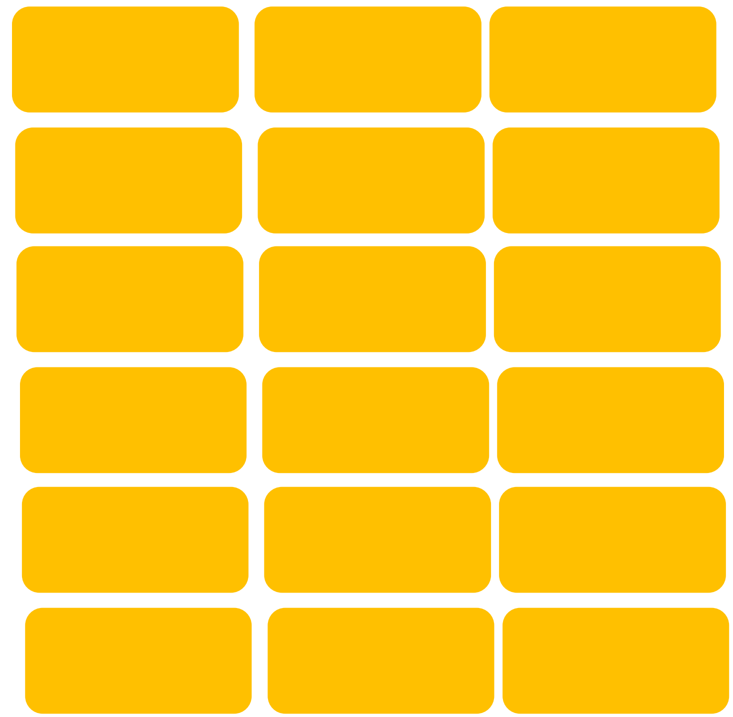
- Linear Programming (LP): f and g_i are linear
- Nonlinear Programming (NLP): some function f, g_i are nonlinear
- Continuous optimization: f and g_i are continuous on an open set containing X, X is closed and convex
- Integer Programming (IP): $X \subseteq \{0,1\}^n$ or $X \subseteq Z^n$
- Constrained optimization: $m > 0$, $X \subset R^n$
- Unconstrained optimization: $m = 0$, $X = R^n$

Applications

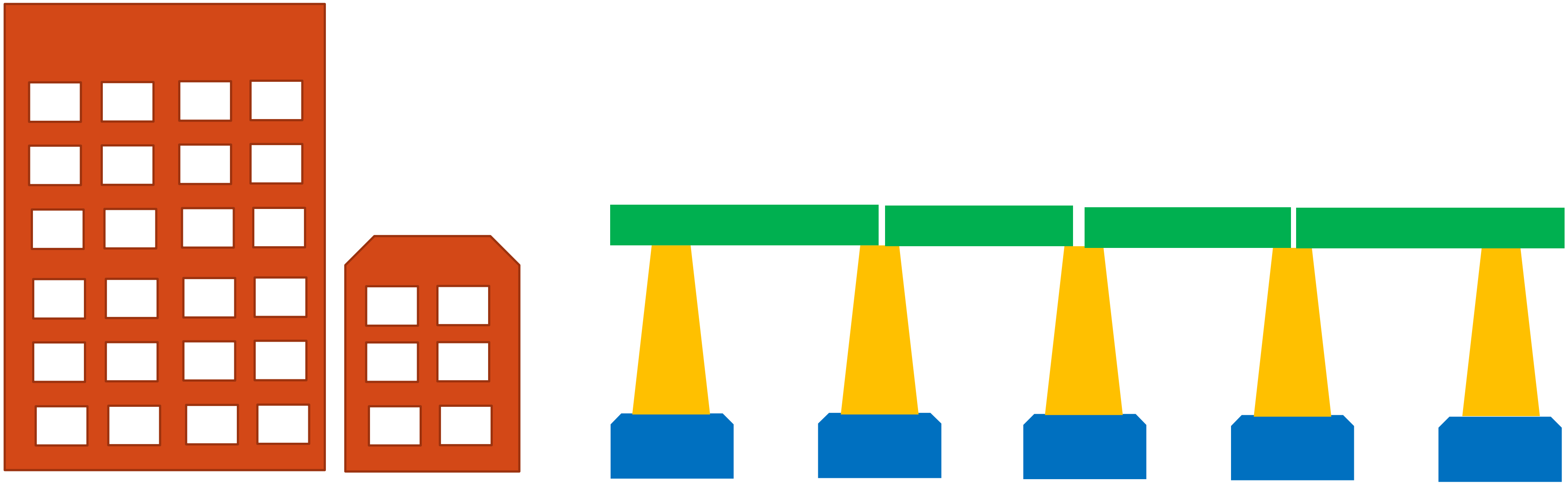
- Production Planning
- Routing in transportation
- Scheduling
- Assignment
- Packing
- Time Tabling
- Network designs
- Machine learning

Agriculture Production Planning

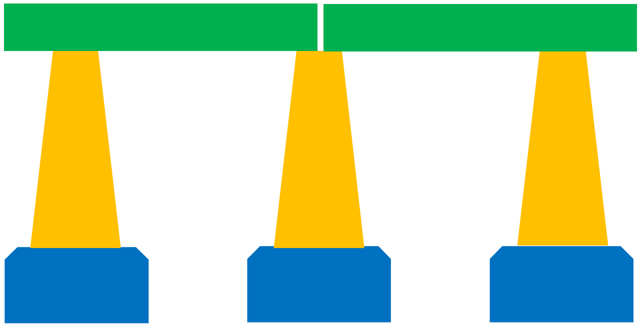
SKU	Chart	Demand
		10000
		25000
		32000
		42500



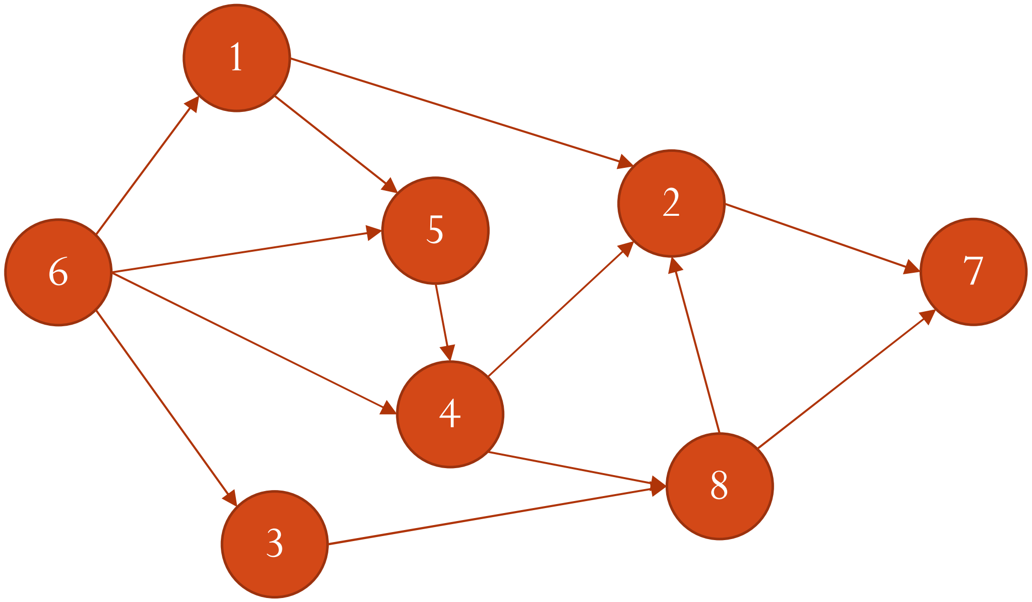
Construction Planning



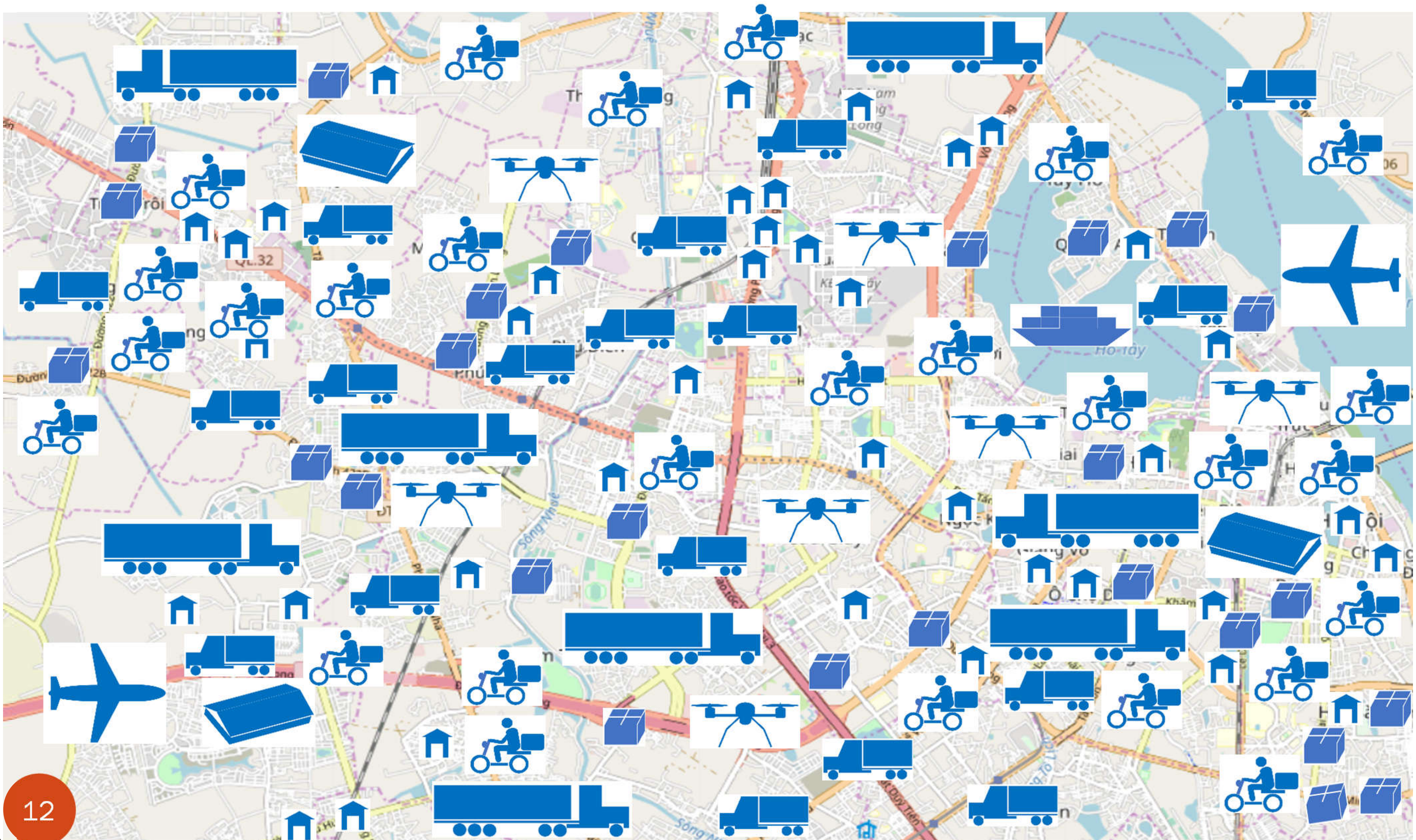
Planning



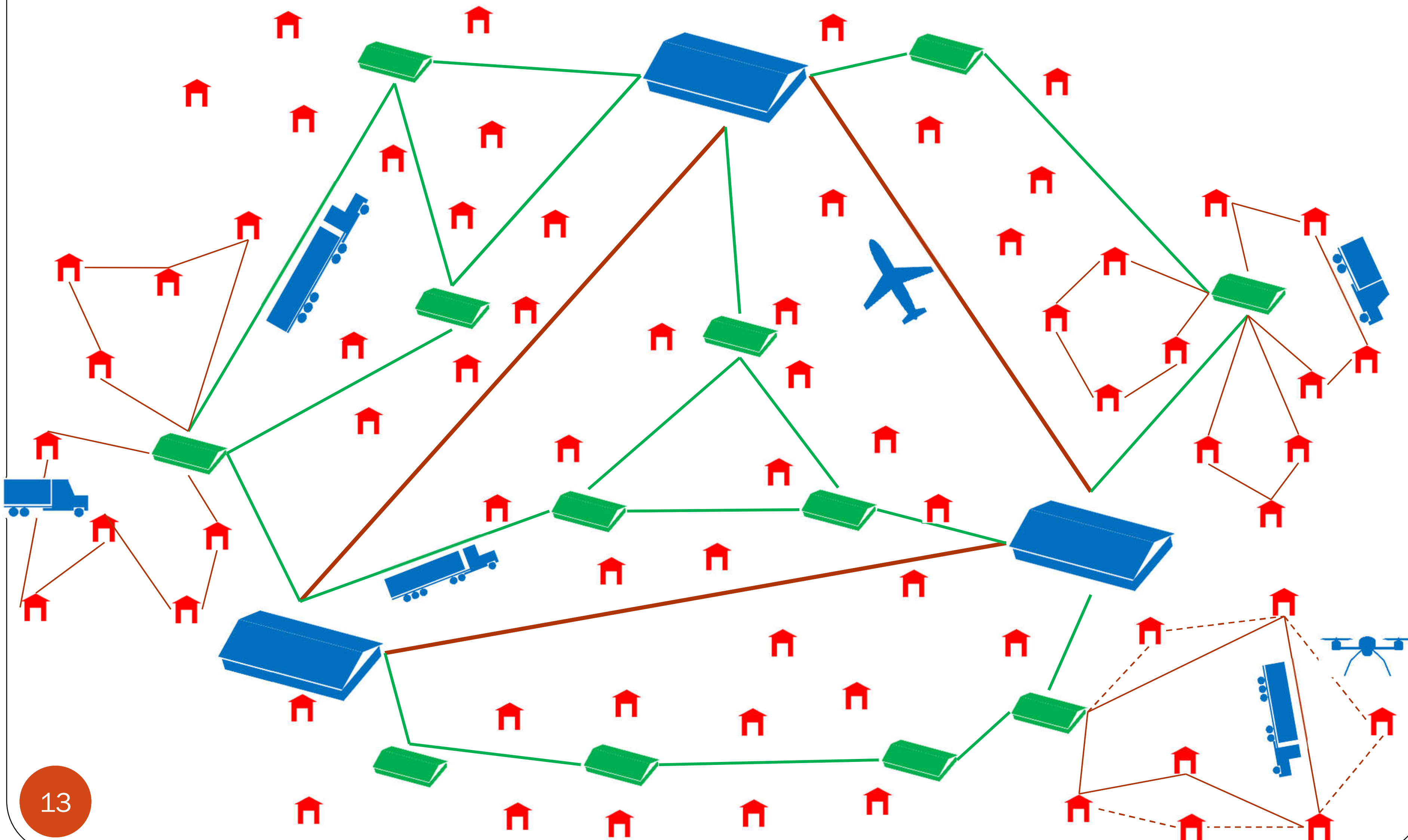
Task	Duration	Predecessors
1	30	6
2	20	1,4,8
3	15	6
4	25	5,6
5	20	1,6
6	45	
7	40	2,8
8	30	3,4



Logistics & Transportation

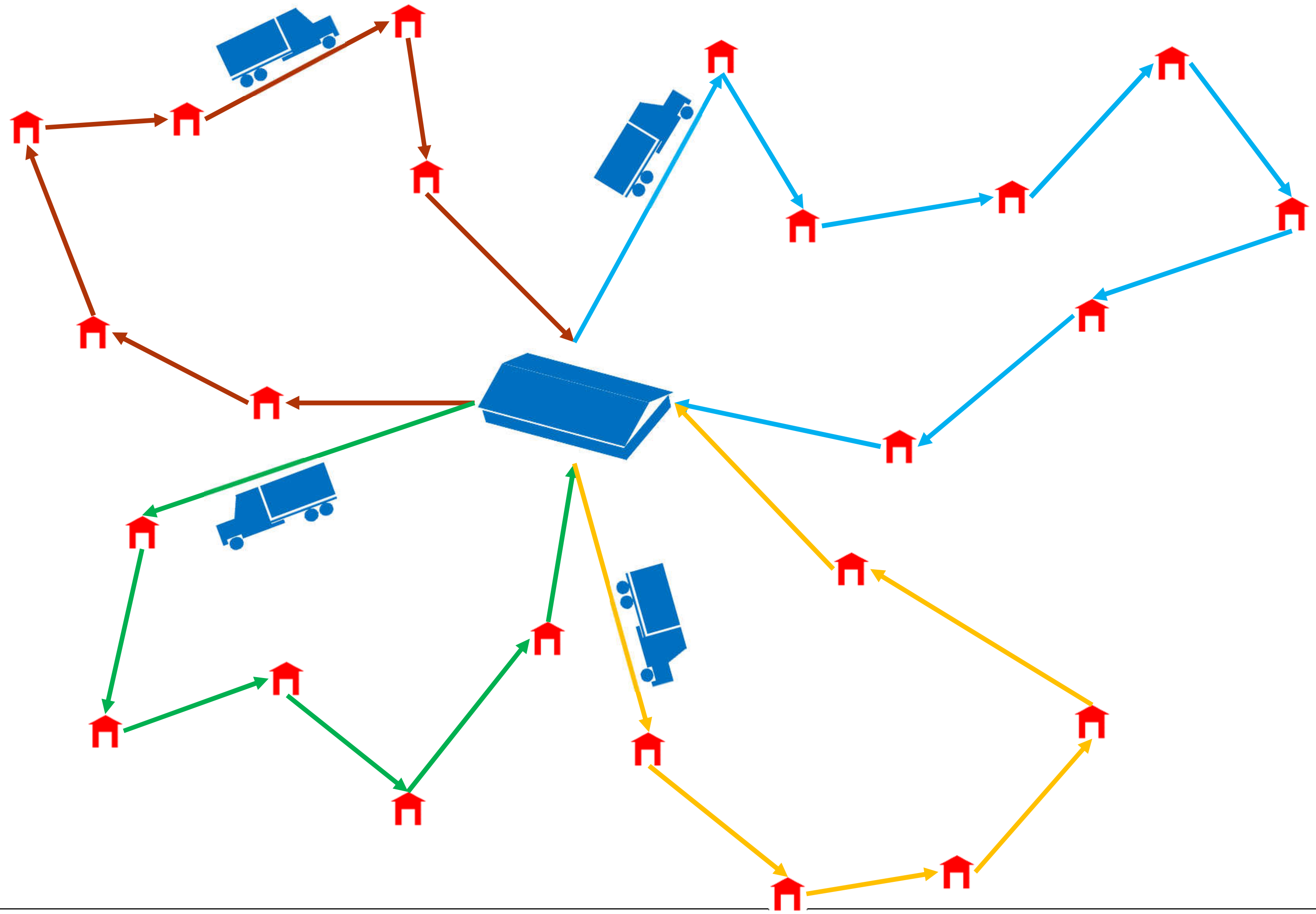


Logistics & Transportation



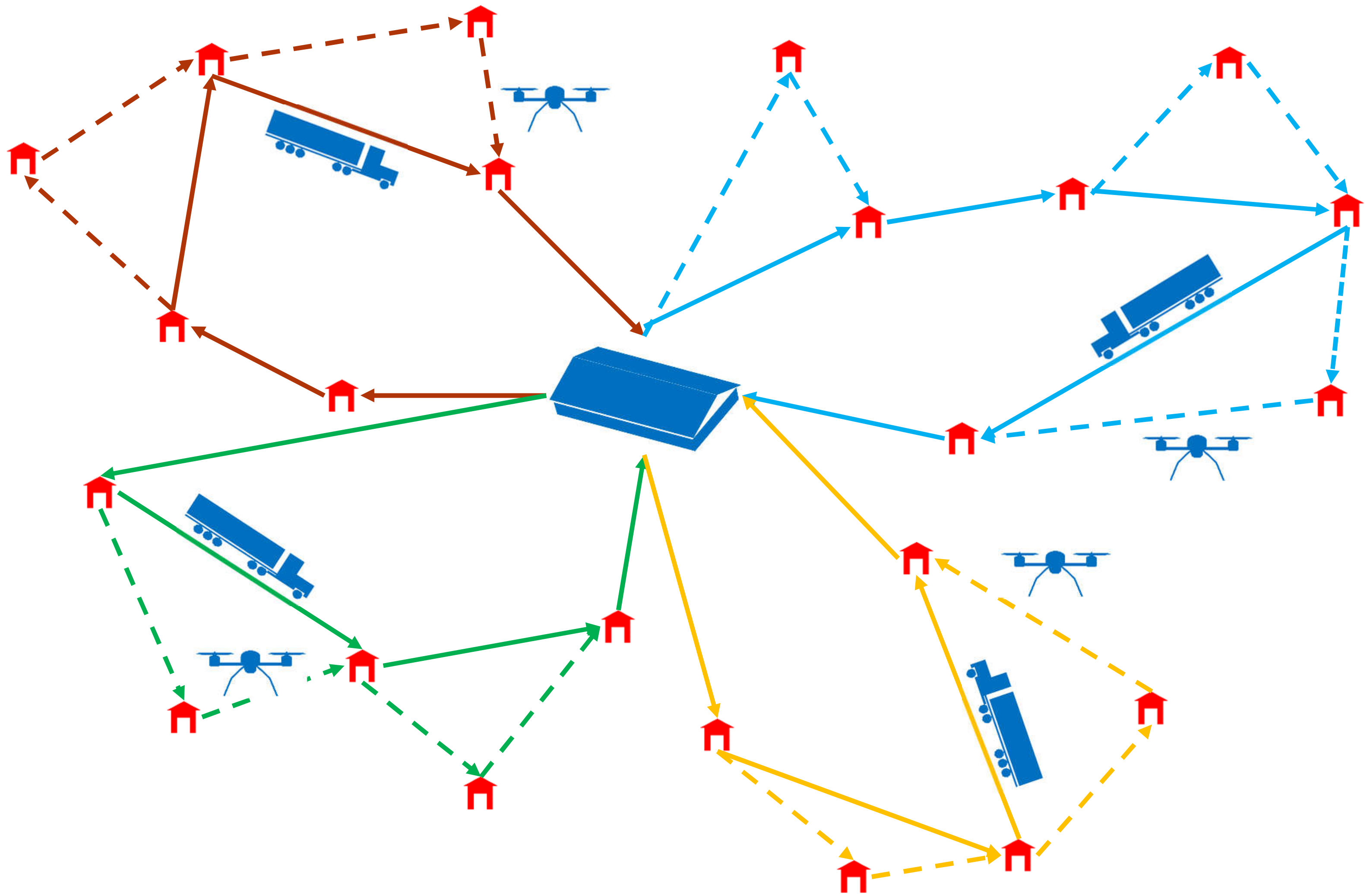
Logistics & Transportation

- How to make a plan for delivering goods to customers



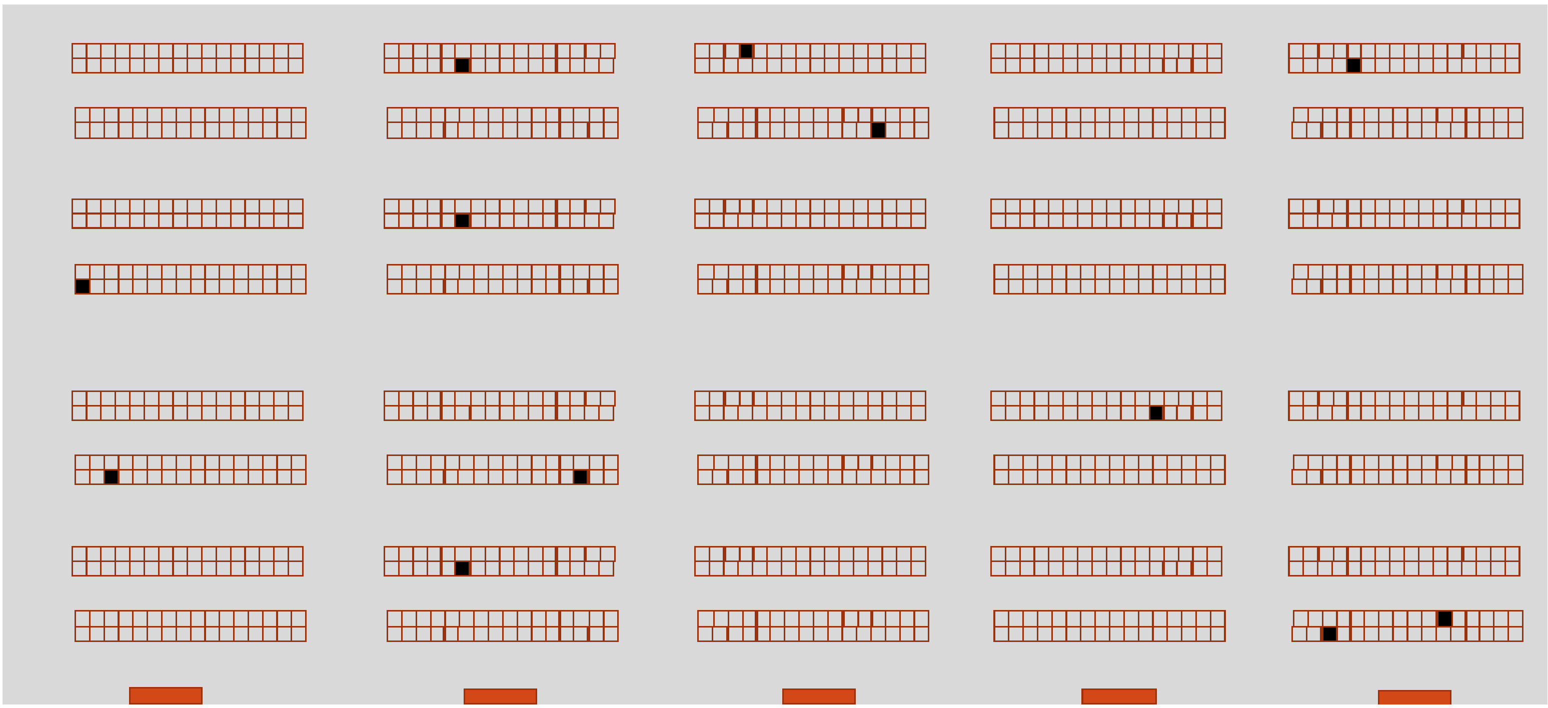
Logistics & Transportation

- How to make a plan for delivering goods to customers



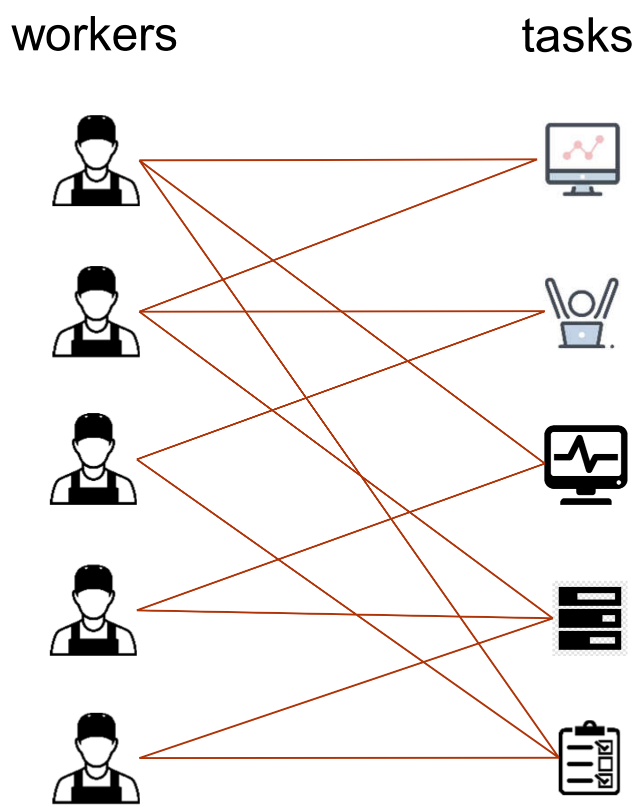
Logistics

- How to pick items in a very large warehouse?



Assignment

- How to assign tasks to workers in an optimal way



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

Time tabling

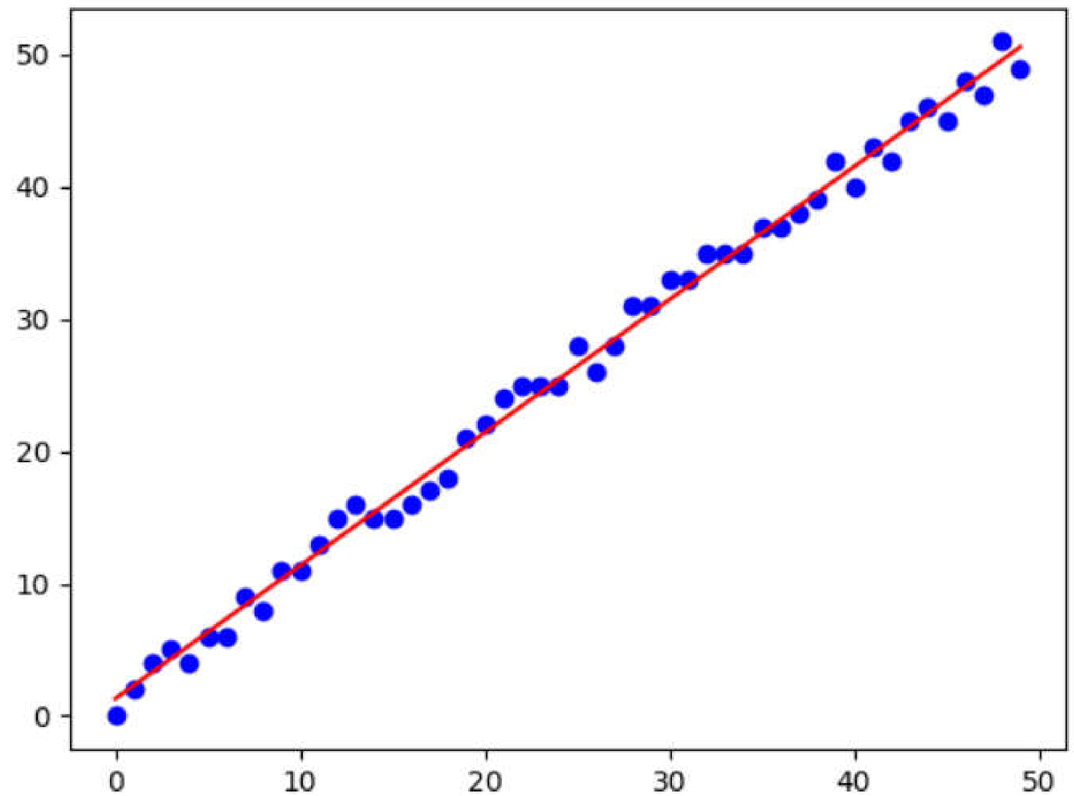
- How to assign classes into slots of the timetable

<table>
<tr><th>Monday</th><th>Tuesday</th><th>Wednesday</th><th>Thursday</th><th>Friday</th></tr>
<tr><td>Data structure & Algorithms, TC-305</td><td rowspan="2">Python Programmin g, D9-302</td><td rowspan="3">Statistics, B1-203</td><td>Technical writing, B1-202</td><td rowspan="2">Networkings , B1-404</td></tr>
<tr><td rowspan="3">Fundamenta l of optimization, B1-402</td><td rowspan="2">Java advanced, B1-204</td></tr>
<tr><td rowspan="2">Machine learning, D6-302</td><td rowspan="2">Image processing, D6-303</td></tr>
<tr><td>Software engineering, D5-102</td><td>Operating systems, D9-101</td></tr>
</table>

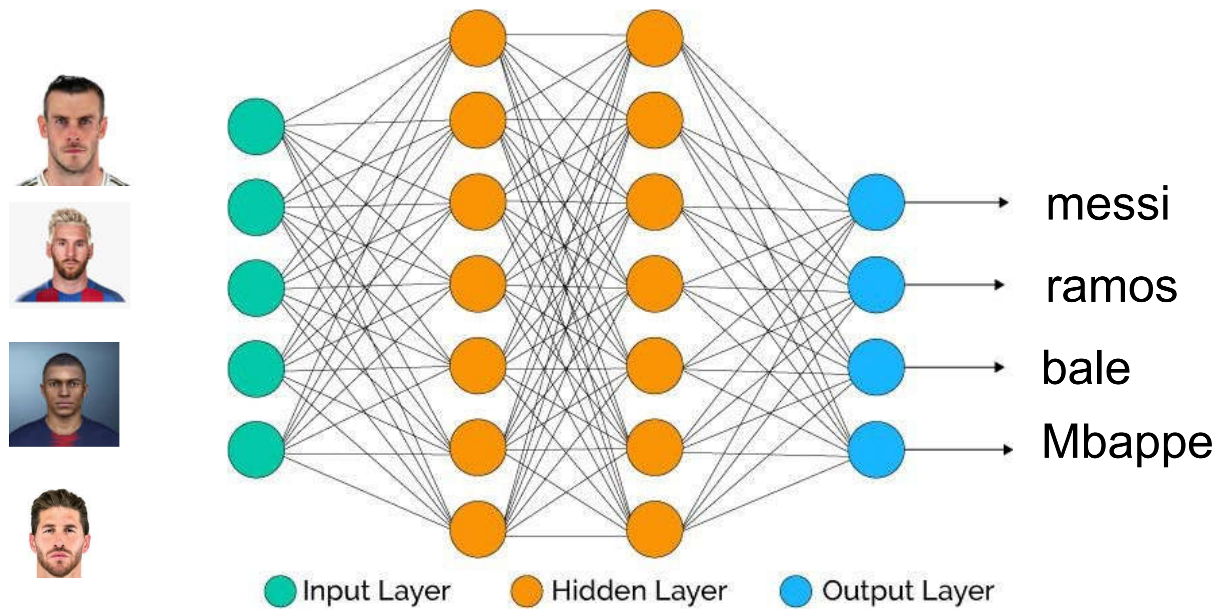
Machine learning

- Prediction

X	Y
43	45
44	46
45	45
46	48
47	47
48	51
49	49
50	?



Computer vision



Demo example with or-tools

$f(x) = 2x_1 + 4x_2 - x_3 \rightarrow \min$

$$4x_1 - x_2 + 2x_3 \leq 7$$

$$x_1 + x_2 + x_3 = 5$$

$$3x_1 + x_2 - 2x_3 \leq 10$$

$$x_1, x_2 \in R, x_1, x_2 \geq 2,$$

$$x_3 \in Z, 0 \leq x_3 \leq 10$$

Demo example with or-tools

```
from ortools.linear_solver import pywraplp

solver = pywraplp.Solver.CreateSolver('DEMO','CBC')
INF = solver.infinity()
x1 = solver.NumVar(2, INF, 'x1')
x2 = solver.NumVar(2, INF, 'x2')
x3 = solver.IntVar(0, 10, 'x3')

c1 = solver.Constraint(-INF, 7)
c1.SetCoefficient(x1,4)
c1.SetCoefficient(x2,-1)
c1.SetCoefficient(x3,2)

c2 = solver.Constraint(5,5)
c2.SetCoefficient(x1,1)
c2.SetCoefficient(x2,1)
c2.SetCoefficient(x3,1)
```

Demo example with or-tools

```
c3 = solver.Constraint(-INF, 10)
c3.SetCoefficient(x1,3)
c3.SetCoefficient(x2,1)
c3.SetCoefficient(x3,-2)

obj = solver.Objective()
obj.SetCoefficient(x1,2)
obj.SetCoefficient(x2,4)
obj.SetCoefficient(x3,-1)

result_status = solver.Solve()
assert result_status == pywraplp.Solver.OPTIMAL
print('Optimal objective value = %f' % solver.Objective().Value())
print('x1 = ',x1.solution_value(),'x2 = ',x2.solution_value(),'x3 = 
',x3.solution_value())
```