

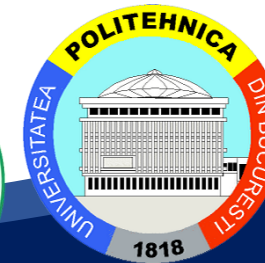


Co-funded by the
Erasmus+ Programme
of the European Union



Advanced Optimization: Techniques and Industrial Applications

Module 3: Optimization and Its Applications in Industry 4 Era



Curriculum Development
of Master's Degree Program in
Industrial Engineering for Thailand Sustainable Smart Industry

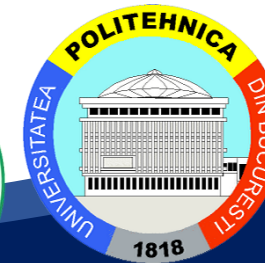


Co-funded by the
Erasmus+ Programme
of the European Union



Session 3.3:

Optimization (Opt) Design in I4.0



Curriculum Development
of Master's Degree Program in
Industrial Engineering for Thailand Sustainable Smart Industry



3.3 Optimization (Opt) Design in Industry 4.0

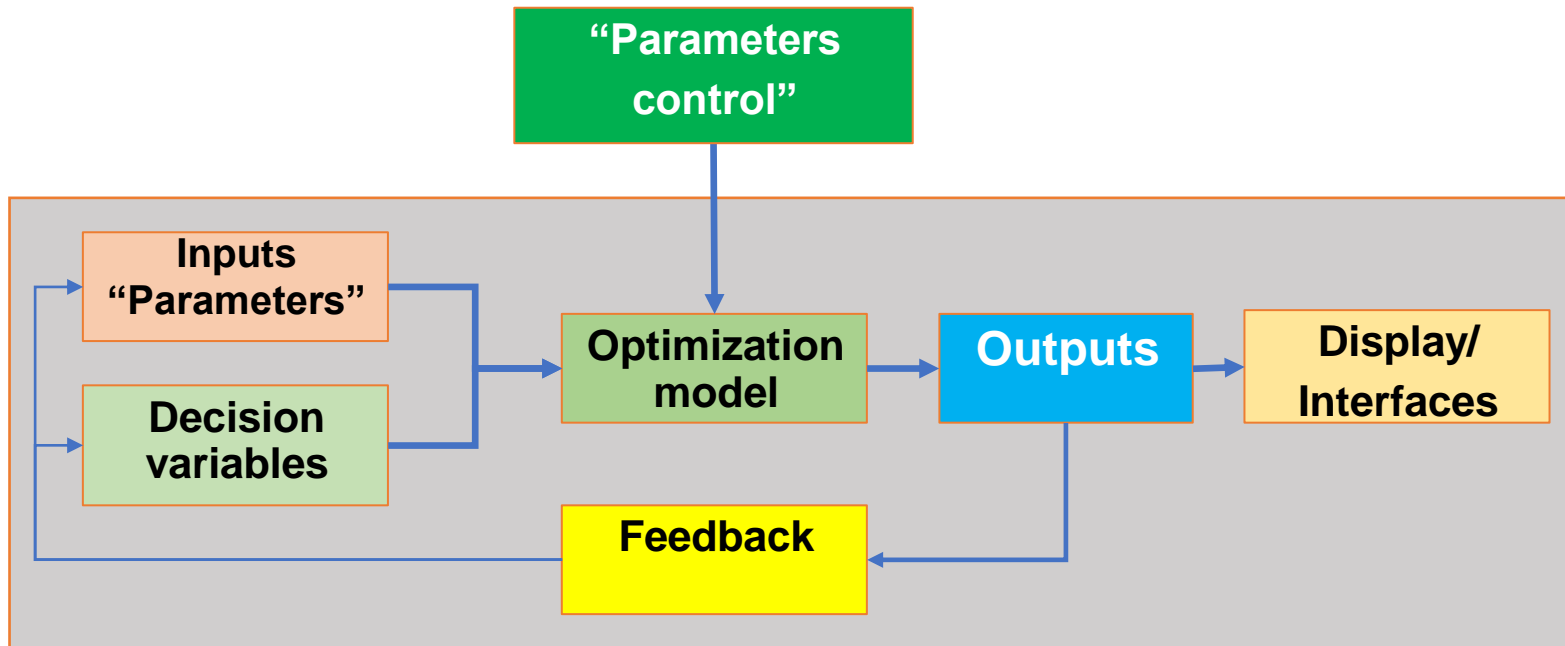
3.3.1 System analysis concept

- 3.3.1 System analysis concept
- 3.3.2 System architecture, module and component design
- 3.3.3 Data input/output user interface design
- 3.3.4 Optimization programming, modeling, and simulation
- 3.3.5 Evaluating the designed system



3.3 Optimization (Opt) Design in Industry 4.0

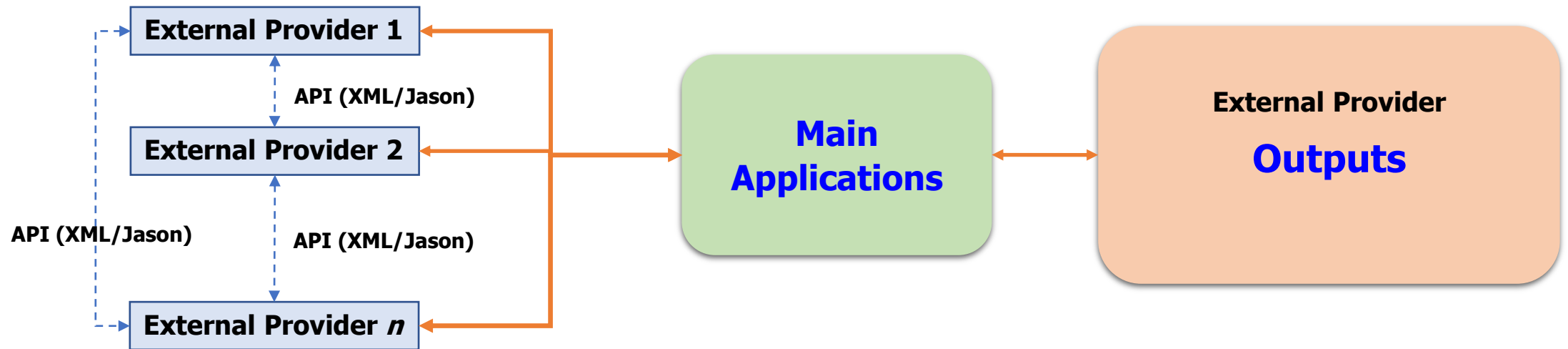
3.3.1 System analysis concept



- ✓ A real-time data collection leads to the issue of handling and analyzing huge data

- **Sources of inputs:**
 - Imported files
 - Mobile
 - Google Cloud
 - Sensors/devices
 - Camera
 - Connected devices
- **Kinds of outputs:**
 - Objective function
 - Decision variables
- **Interfaces:**
 - Reports
 - Graphs & Tables
 - Dashboards

3.3.2 System architecture, module and component design



- External Providers = Sources of inputs

- Imported files
- Mobile
- Google Cloud
- Sensors/devices
- Camera
- Connected devices
- RFID

- Algorithms
- Metaheuristics
- Software

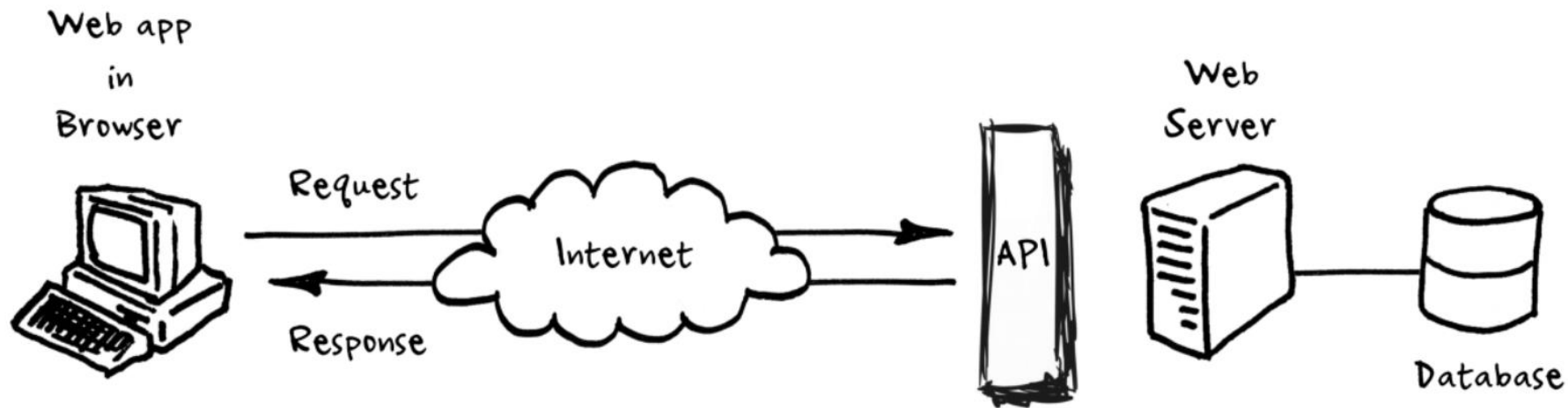
Report / Visualization

- Graphs
- Tables
- Figures
- Dashboards

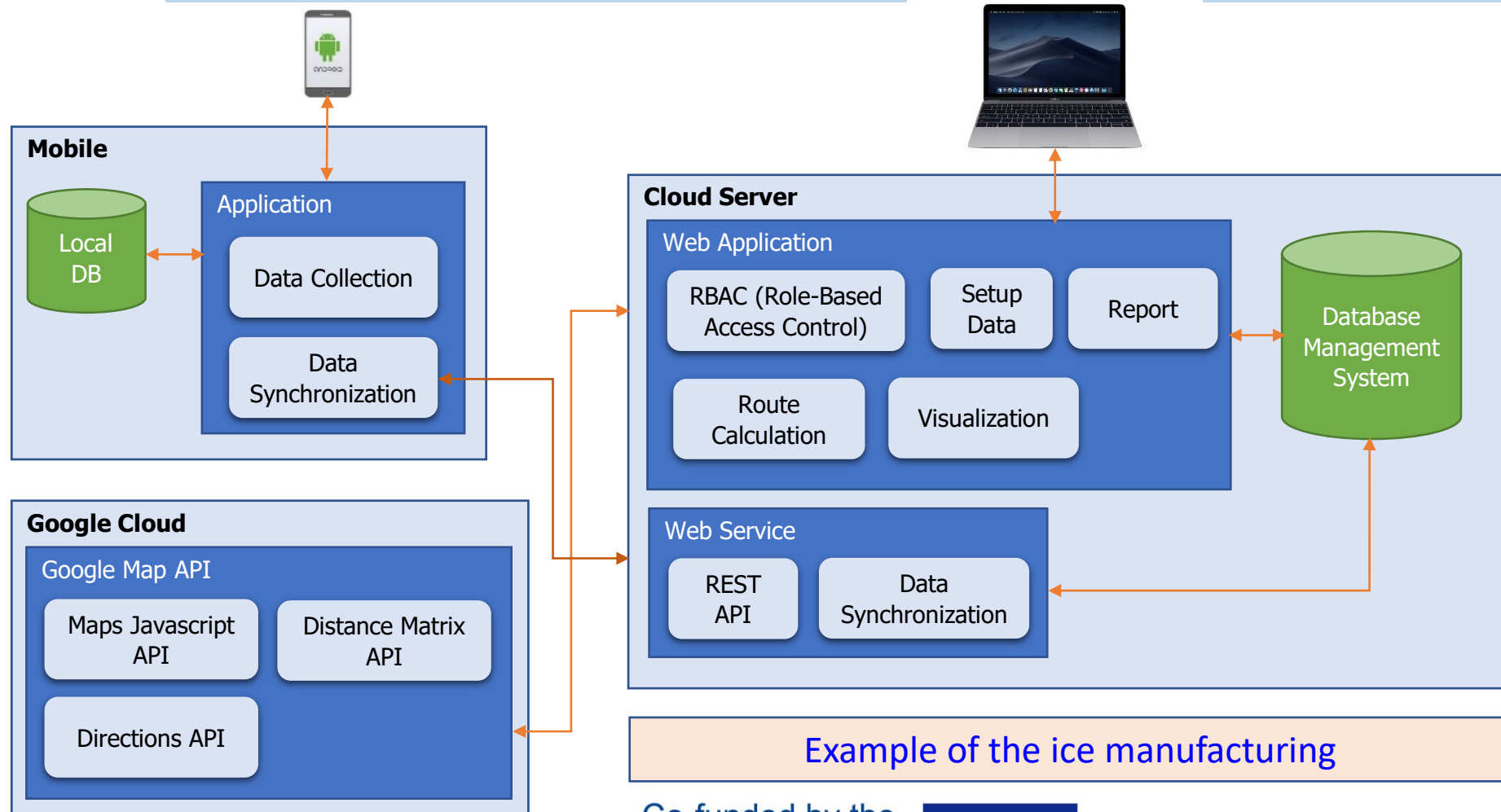
Application Programming Interface (API)

- An **API** is a set of **programming code that enables data transmission between one software product and another**. It also contains the terms of this data exchange. (altexsoft.com)
- API **allows an application to communicate with another application**, or an operating system, database, network, etc. (Eric Conrad, [Seth Misener](#), Joshua Feldman, [CISSP Study Guide \(Third Edition\)](#), 2016)

Example



3.3.2 System architecture, module and component design





3.3.2 System architecture, module and component design

- ✓ **Web Application** is the program that users (i.e., route planners) can operate through a Browser such as Chrome or Safari.
- ✓ In this study, the Web Application developed is comprised of the relationships and interactions between application components such as:
 - **Role-based access control (RBAC)** to provide security, because it only allows employees and drivers to access the information they need to do their jobs, while preventing them from accessing additional information that is not relevant to them (Techpedia, 2019).
 - **Setup Data**
 - **Reporting of sales volumes and travel distance of each vehicle**
 - **Route Calculation** for each vehicle
 - **Visualization to display the route obtained from the calculation.** Route maps and route flow are displayed. Additionally, Route Calculation and Visualization will work through the Google Cloud system with 3 functions:
 - Map Javascript API for displaying maps
 - Distance Matrix API for calculating travel time and distance
 - Directions API for calculating vehicle route
- ✓ The **Web Service** is a collection of open protocols and standards used for exchanging data between applications.



3.3.3 Data input/output user interface design

- Determine types of inputs: For example, the inputs of the ice transportation model:

- Maximum distance travel
- Distance from one point to another
- Fuel consumption of a vehicle
- Driver ID
- Customer demand
- Customer info (location, delivery conditions, etc...)

- What are the inputs related to the optimization model?
- How to get the inputs?
- What are the outputs?
- How to display the inputs/outputs?

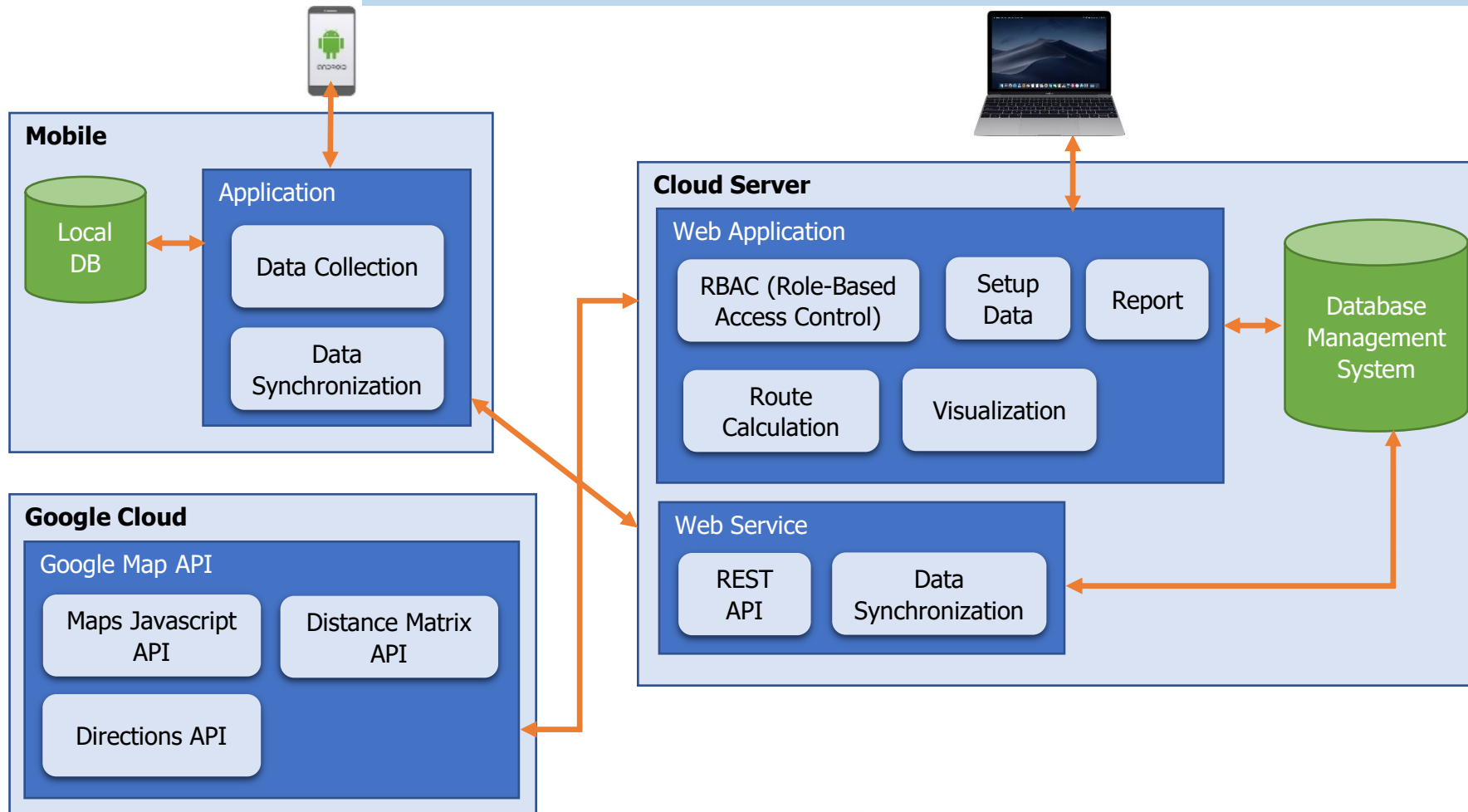
- Determine sources of inputs:

- Imported files or local database
- Mobile
- Google Cloud
- Sensors
- Camera
- Connected devices
- RFID

- Determine kinds of outputs and platforms:

- Objective function
- Decision variables
- Display of the outputs:
- GUI:
 - ✓ Reports
 - ✓ Graphs & Tables
 - ✓ Dashboard

3.3.3 Data input/output user interface design



- Sources of inputs:

- Imported files
- Mobile
- Google Cloud
- Sensors
- Camera
- Connected devices

- Kinds of outputs:

- Obj function
- Decision variables

- GUI:

- Reports
- Graphs & Tables

3.3.3 Data input/output user interface design

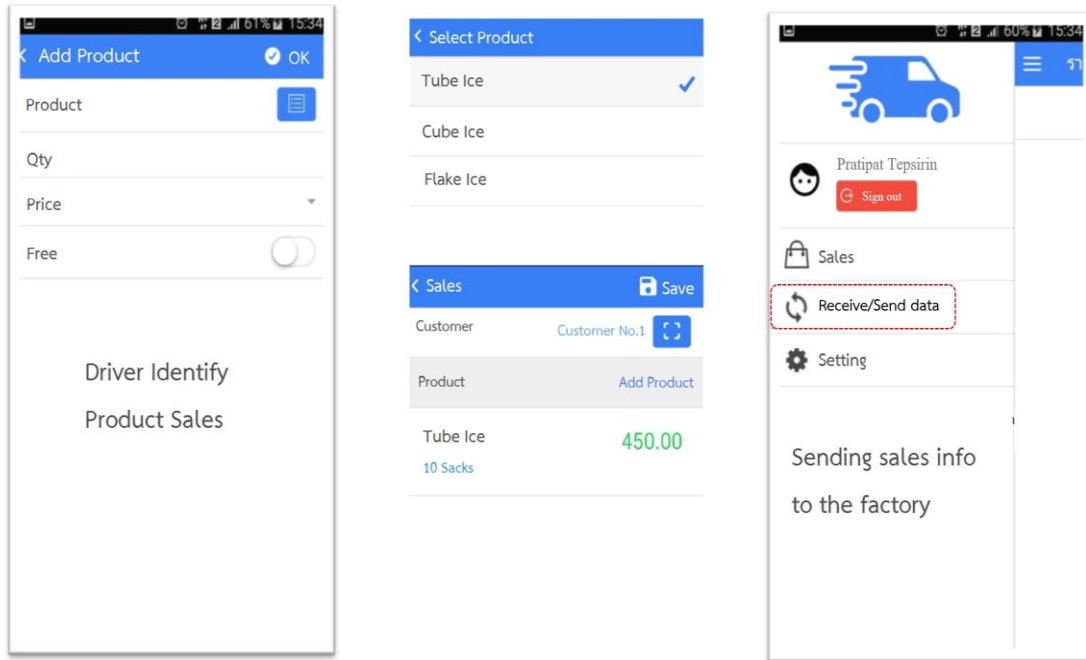


Fig. 5 (a) Mobile application for a driver to obtain the orders confirmed by the customers

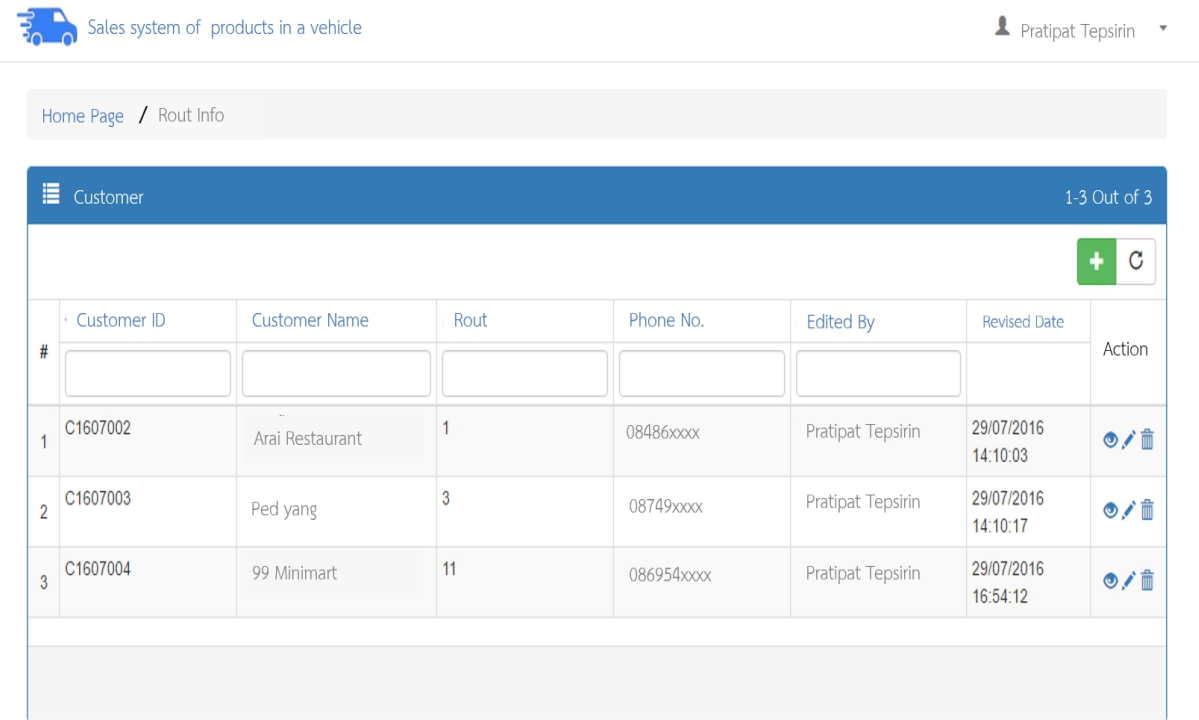

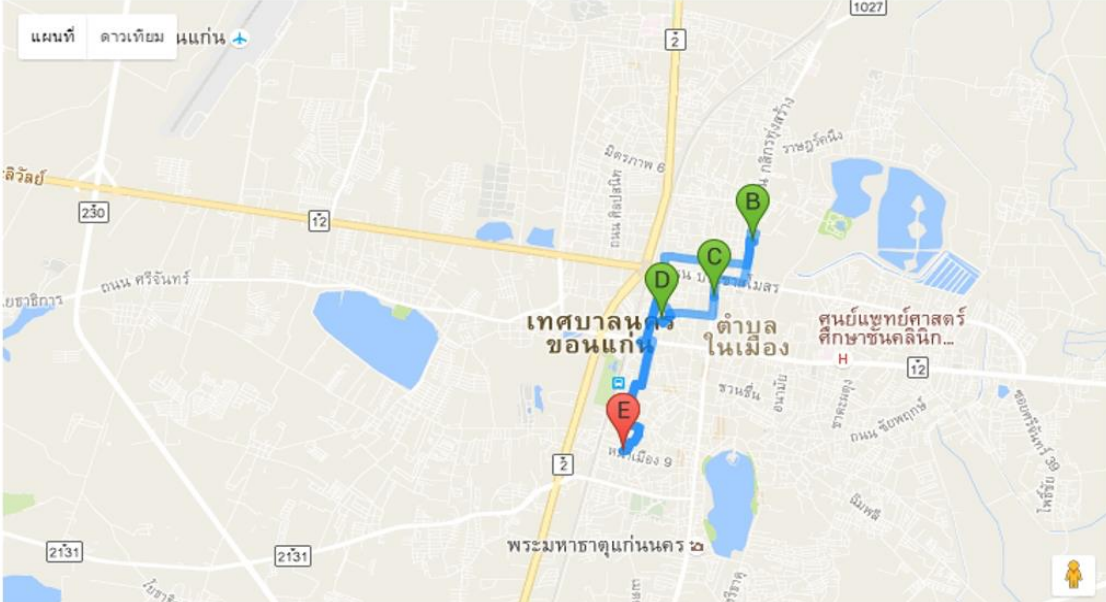


Fig. 5 (b) Mobile application for a driver to obtain the orders confirmed by the customers

3.3.3 Data input/output user interface design

 Sales system of products in a vehicle Pratipat Tepsirin

Home Page / Route info



แผนที่ ดาวเทียม นกแก้ว

↑ LS1

- ↑ Head east toward wuttaram8 road
⚠ Restricted usage road
29 m
- ↘ Turn right onto wuttaram8 road
43 m
- ↘ Turn right onto namaung road
300 m
- ↘ Turn right onto darunsamran road
850 m
- 📍 At the roundabout, take the 1st exit and stay on darunsamran road

Fig. 6 Visualization to display the route in terms of route maps and route flow of the mobile application

3.3.4 Optimization programming, modeling, and simulation

Algorithm 1: Combination of Particle Swam Optimization and Adaptive Large Neighborhood Search

```

Input : Problem size, Population size
Output:  $P_{n,best}$ 
Set Population equal to 1 set  $P_{n,best} \leftarrow$  great number
For  $i=1$  until  $i=$ Population size
     $P_{velocity} \leftarrow$  random real number
     $P_{position} \leftarrow$  random real number
    Set  $P_{n,best}$  equal to  $P_{position}$ 
    Perform Adaptive Large neighborhood search (ALNS)
    While (termination condition of ALNS is not met)
        Select Destroy method (4 destroy method, ProbDesMethod)
        Select Repair method (2 repair method, ProbReMethod)
        Update Score of destroy and repair method
    End
    If ( $Cost(P_{n,best}) \leq Cost(P_{repair})$ )
         $P_{n,best} \leftarrow P_{repair}$ 
    End
End
While (Stopping condition does not met)
    For ( $P \in$  Population)
         $P_{velocity} \leftarrow UpdateVelocity(P_{velocity}, P_{n,best}, P_{n,best})$ 
         $P_{position} \leftarrow UpdatePosition(P_{position}, P_{velocity})$ 
        Perform Adaptive Large neighborhood search (ALNS)
        While(termination condition of ALNS is not met)
            Select Destroy method (6 destroy method, ProbDesMethod)
            Select Repair method (3 repair method, ProbReMethod)
            Update Score of destroy/repair methods
        End
        If ( $Cost(P_{position}) \leq Cost(P_{n,best})$ )
             $P_{n,best} \leftarrow P_{position}$ 
        If ( $Cost(P_{n,best}) \leq Cost(P_{repair})$ )
             $P_{n,best} \leftarrow P_{repair}$ 
        End
    End
End
End
    
```

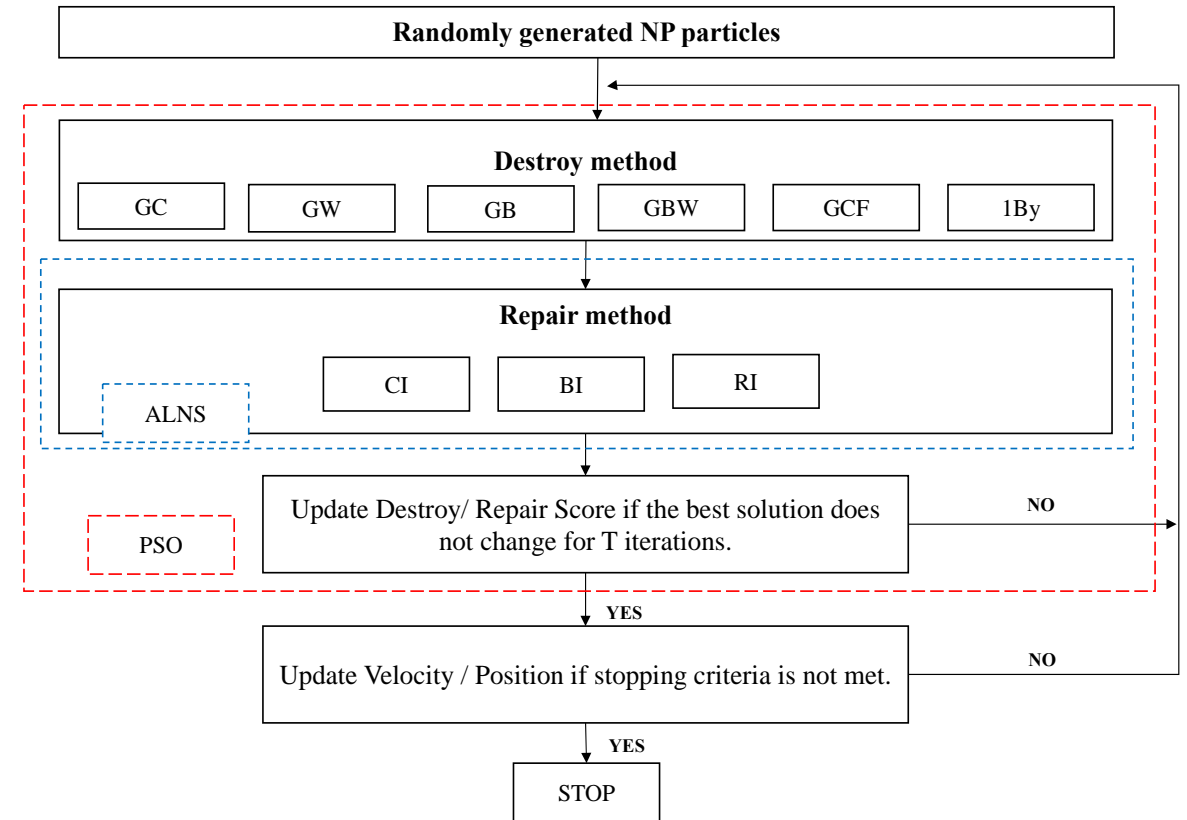


Fig. 2. Framework of the proposed Method



Group Assignment 2

Each project group, must complete the following issues for one of the 2 problems:

- System analysis concept
- System architecture, module and component design
- Data input/output user interface design
- 2 problems
 - ✓ **Problem 1:** Routing of university's buses (Hint: Be able to apply a camera to determine the number of customers waiting at each bus stop)
 - ✓ **Problem 2:** Hotel Maid scheduling (Hint: Be able to apply QR code to determine the number of rooms to be serviced at each time period)





Group Assignment 2

Problem 1: Routing of university's buses (Apply a camera to determine the number of customers waiting at each bus stop)

Problem Statement:

In Thai universities, transportation services are provided for customers (i.e., students and staff) which mainly focus on providing safety, reliability and a cost efficient and environmentally friendly solution. It is increasingly important to today's workforce. Currently, the buses are scheduled in a fixed route with fixed timetable in order to attract students and staff. The services start from 6:00 am. until 20:00 pm. To start the service, a fleet of heterogeneous vehicles located at a depot must be routed to pick up customers from the bus stops scattered geographically. The number of customers at each bus stop may be different and varies by time. However, at some bus stops, there are no customers at some periods of times. This cause the university pays very high cost for this service since all buses are scheduled to service on their route.

Since the number of customers varies with time, and there are a lot of existing bus stops on the routes with different number of waiting customers, bus routes and schedules are required. Features of the university's bus transportation problem is shown in Fig. xx.

Objective function: To minimize the number of buses used at each time period



Illustration of the problem

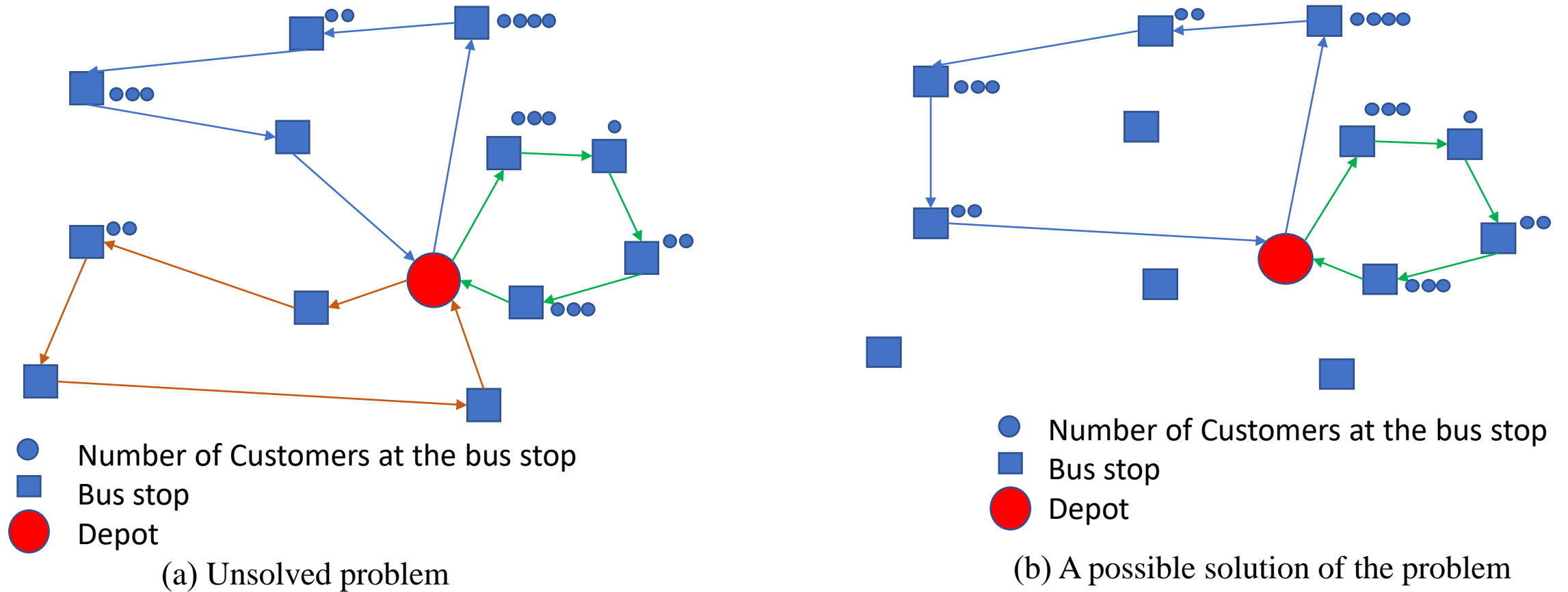


Fig. 1. Features of the university transportation problem





Group Assignment 2 (cont'd)

Problem 2: Hotel Maid scheduling (Apply QR code to determine the number of rooms to be serviced at each time period)

Problem statement:

Currently, room cleaning service is provided to attract customers for the hotel guests. To increase the customer satisfaction, the hotel maids must be available all times. The challenge of today's service is that the maids do not know exactly which room is needed for the service by which time. They have to check the sign hanged in front of the room for every room. This cause the delay in service and high cost of hiring maids. In order to to speed up processes and enhance customer satisfaction while increasing agility, maid scheduling to clean up the room is required. However, the requests from each room must be known and accurate. The QR code is created for that purpose. Each customer who require the room cleaning must scan the QR code to notify which time slot the room is cleaned. All data must be processed in the hotel database. The maid scheduling is then determined.

Objective function: To minimize the number of hotel maids required



3.3.5 Evaluating the designed system

Introduction

- Most of input data are estimated values, and hence, would be updated when new information becomes available. Some constraints or variables may be missed at the beginning and needed to be incorporated later also.
- When updating input data, it is not a good approach to solve the problem from the beginning because this will be time-consuming for practical problems with large sizes
- In this part, we will consider the use of the *optimal* basis **B** to find the new optimal solution for a linear program when some of the problem data are changed *without* resolving the problem from the beginning.



Consider the standard form of LP:

$$\begin{array}{ll} \text{Minimize} & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} & \mathbf{Ax} = \mathbf{b} \geq \mathbf{0} \\ & \mathbf{x} \geq \mathbf{0} \end{array}$$

The following data variations will be taken into consideration:

- Change in the right-hand side vector \mathbf{b} .
- Change in the coefficient vector of the objective function \mathbf{c}
- Change in the constraint matrix \mathbf{A}
- Addition of a new variable
- Addition of a new constraint

Introduction

For simplicity, the matrix form of the simplex tableau will be employed

	Z	\mathbf{x}_D	\mathbf{x}_B	RHS
Z	1	$\mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{D} - \mathbf{c}_D^T$	$\mathbf{0}^T$	$\mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b}$
\mathbf{x}_B	$\mathbf{0}$	$\mathbf{B}^{-1} \mathbf{D}$	\mathbf{I}	$\mathbf{B}^{-1} \mathbf{b}$

In which:

- $\mathbf{x}_B, \mathbf{x}_D$: vectors of basic and nonbasic variables.
- $\mathbf{c}_B, \mathbf{c}_D$: coefficient vectors of the objective function associated with basic and nonbasic variables.
- \mathbf{b} : right-hand side vector.
- \mathbf{D} : part of matrix \mathbf{A} associated with nonbasic variables.

Example 1:

$$\text{Min } -2x_1 - x_2$$

s.t

$$x_1 + \frac{8}{3}x_2 \leq 4$$

$$x_1 + x_2 \leq 2$$

$$2x_1 \leq 3$$

$$x_1, x_2 \geq 0$$

$$\text{Min } Z = -2x_1 - x_2$$

s.t

$$x_1 + \frac{8}{3}x_2 + x_3 = 4$$

$$x_1 + x_2 + x_4 = 2$$

$$2x_1 + x_5 = 3$$

$$x_1, x_2 \geq 0$$

\Leftrightarrow

We have:

$$\mathbf{A} = \begin{bmatrix} 1 & 8/3 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} -2 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 4 \\ 2 \\ 3 \end{bmatrix}$$

Consider a simplex tableau of the problem:

	Z	x_1	x_2	x_3	x_4	x_5	RHS
Z	1	0	1	0	0	-1	-3
x_3	0	0	8/3	1	0	-1/2	5/2
x_4	0	0	1*	0	1	-1/2	1/2
x_1	0	1	0	0	0	1/2	3/2

Introduction

$$\bullet \mathbf{x}_B = \begin{bmatrix} x_3 \\ x_4 \\ x_1 \end{bmatrix}, \quad \mathbf{x}_D = \begin{bmatrix} x_2 \\ x_5 \end{bmatrix} \Rightarrow \mathbf{c}_B = \begin{bmatrix} 0 \\ 0 \\ -2 \end{bmatrix}, \quad \mathbf{c}_D = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

$$\bullet \mathbf{B} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix} \Rightarrow \mathbf{B}^{-1} = \begin{bmatrix} 1 & 0 & -1/2 \\ 0 & 1 & -1/2 \\ 0 & 0 & 1/2 \end{bmatrix}$$

$$\bullet \mathbf{D} = \begin{bmatrix} 8/3 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$



Hence

$$\bullet \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{D} - \mathbf{c}_D^T = [0 \ 0 \ -2] \begin{bmatrix} 1 & 0 & -1/2 \\ 0 & 1 & -1/2 \\ 0 & 0 & 1/2 \end{bmatrix} \begin{bmatrix} 8/3 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} - [-1 \ 0] \quad \bullet \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b} = [0 \ 0 \ -2] \begin{bmatrix} 1 & 0 & -1/2 \\ 0 & 1 & -1/2 \\ 0 & 0 & 1/2 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 3 \end{bmatrix} = -3$$

$$= [0 \ -1] - [-1 \ 0] = [1 \ -1].$$

$$\bullet \mathbf{B}^{-1} \mathbf{D} = \begin{bmatrix} 1 & 0 & -1/2 \\ 0 & 1 & -1/2 \\ 0 & 0 & 1/2 \end{bmatrix} \begin{bmatrix} 8/3 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 8/3 & -1/2 \\ 1 & -1/2 \\ 0 & 1/2 \end{bmatrix} \quad \bullet \mathbf{B}^{-1} \mathbf{b} = \begin{bmatrix} 1 & 0 & -1/2 \\ 0 & 1 & -1/2 \\ 0 & 0 & 1/2 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 5/2 \\ 1/2 \\ 3/2 \end{bmatrix}.$$

Note:

If the initial basic solution can be formed by use of slack variables then the part of matrix **A** associated with the slack variables in any iteration of the simplex method will define the inverse \mathbf{B}^{-1} of the current basis **B**



Change in the right-hand side vector: $b \Rightarrow b'$

This will affect RHS values of the simplex tableau. There are two cases:

Case 1: $\mathbf{B}^{-1}\mathbf{b}' \geq \mathbf{0}$: the current basis \mathbf{B} is still optimal and the new optimal objective value is: $\mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b}'$.

Example 2:

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


$b \Rightarrow b'$



Change in the right-hand side vector: $b \Rightarrow b'$

The optimal simplex tableau:

	Z	x_1	x_2	x_3	x_4	x_5	RHS
Z	1	0	-3	-1	-2	0	-12
x_1	0	1	1	1	1	0	6
x_5	0	0	3	1	1	1	10





Change in the right-hand side vector: $b \Rightarrow b'$

Change the right-hand side vector to $\mathbf{b}' = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$. We have:

$\mathbf{B}^{-1}\mathbf{b}' = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 3 \\ 7 \end{bmatrix} \geq \mathbf{0} \Rightarrow$ still optimal with new optimal objective value:

$$\mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b}' = \begin{bmatrix} -2 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} -2 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 7 \end{bmatrix} = -6$$



$b \Rightarrow b'$



Change in the right-hand side vector: $b \Rightarrow b'$

The new optimal simplex tableau:

	Z	x_1	x_2	x_3	x_4	x_5	RHS
Z	1	0	-3	-1	-2	0	-6
x_1	0	1	1	1	1	0	3
x_5	0	0	3	1	1	1	7



$b \Rightarrow b'$



Change in the right-hand side vector: $b \Rightarrow b'$

Case 2: $\mathbf{B}^{-1}\mathbf{b}' < \mathbf{0}$: the current basis \mathbf{B} is not optimal. Recompute both $\mathbf{B}^{-1}\mathbf{b}'$ and $\mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{b}'$ to form a new simplex tableau and apply the *dual* simplex method to find the new optimal solution





Change in the coefficient vector of the objective function: $c \Rightarrow \bar{c}$

This will affect row 0 of the simplex tableau, i.e., values of $\bar{c}_B^T \mathbf{B}^{-1} \mathbf{D} - \bar{c}_D^T$ and $\bar{c}_B^T \mathbf{B}^{-1} \mathbf{b}$. However, it is noted that only $\bar{c}_B^T \mathbf{B}^{-1} \mathbf{D} - \bar{c}_D^T$ has effect on optimality. There are two cases:

1. $\bar{c}_B^T \mathbf{B}^{-1} \mathbf{D} - \bar{c}_D^T \leq \mathbf{0}^T$: the current basis \mathbf{B} is still optimal and the new (might be not!) optimal objective value is $\bar{c}_B^T \mathbf{B}^{-1} \mathbf{b}$.
2. $\bar{c}_B^T \mathbf{B}^{-1} \mathbf{D} - \bar{c}_D^T \not\leq \mathbf{0}^T$: the current basis \mathbf{B} is not optimal. Recompute both $\bar{c}_B^T \mathbf{B}^{-1} \mathbf{D} - \bar{c}_D^T$ and $\bar{c}_B^T \mathbf{B}^{-1} \mathbf{b}$ to form a new simplex tableau and apply the *primal* simplex method to find the new optimal solution.





Change in the coefficient vector of the objective function: $c \Rightarrow \bar{c}$

Example 3: Reconsider the problem in Example 2

- Replace $c_2 = 1$ by $\bar{c}_2 = -3$, we have:

$$\bar{\mathbf{c}}_B^T \mathbf{B}^{-1} \mathbf{D} - \bar{\mathbf{c}}_D^T = \begin{bmatrix} -2 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 3 & 1 & 1 \end{bmatrix} - \begin{bmatrix} -3 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -2 \end{bmatrix}$$

$$\bar{\mathbf{c}}_B^T \mathbf{B}^{-1} \mathbf{b} = \begin{bmatrix} -2 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 6 \\ 4 \end{bmatrix} = -12$$



b ⇒ b'



Change in the coefficient vector of the objective function: $c \Rightarrow \bar{c}$

The new simplex tableau:

	Z	x_1	x_2	x_3	x_4	x_5	RHS
Z	1	0	1	-1	-2	0	-12
x_1	0	1	1	1	1	0	3
x_5	0	0	3	1	1	1	7

Note: Due to x_2 is not in the current basis \mathbf{B} , the change of c_2 in this case has affects only the coefficient of x_2 in row 0. This fact can be utilized to help simplify the calculation.





Change in the coefficient vector of the objective function: $c \Rightarrow \bar{c}$

The above simplex tableau is not optimal and the primal simplex method can be used to find the new optimal solution as follows:

	Z	x_1	x_2	x_3	x_4	x_5	RHS
Z	1	0	0	-40/3	-70/3	-10/3	-460/3
x_1	0	1	0	20/3	20/3	-10/3	80/3
x_2	0	0	1	10/3	10/3	10/3	100/3



Change in the coefficient vector of the objective function: $c \Rightarrow \bar{c}$

- Replace $c_1 = -2$ by $\bar{c}_1 = 0$, we have:

$$\bar{\mathbf{c}}_B^T \mathbf{B}^{-1} \mathbf{D} - \bar{\mathbf{c}}_D^T = \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 3 & 1 & 1 \end{bmatrix} - \begin{bmatrix} 1 & -1 & 0 \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 \end{bmatrix}$$

$$\bar{\mathbf{c}}_B^T \mathbf{B}^{-1} \mathbf{b} = \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 6 \\ 4 \end{bmatrix} = 0$$



b ⇒ b'



Change in the coefficient vector of the objective function: $c \Rightarrow \bar{c}$

The new simplex tableau:

	Z	x_1	x_2	x_3	x_4	x_5	RHS
Z	1	0	-1	1	0	0	0
x_1	0	1	1	1*	1	0	6
x_5	0	0	3	1	1	1	10

This is not an optimal tableau!



b ⇒ b'



Change in the coefficient vector of the objective function: $c \Rightarrow \bar{c}$

The optimal tableau found by simplex method is as follows:

	Z	x_1	x_2	x_3	x_4	x_5	RHS
Z	1	-1	-2	0	-1	0	-6
x_3	0	1	1	1	1	0	6
x_5	0	-1	2	0	0	1	4





Change in the constraint matrix \mathbf{A} : $\mathbf{A}_j \Rightarrow \bar{\mathbf{A}}_j$

There are two cases that might happen:

Case 1: \mathbf{A}_j is not a basic column ($\mathbf{D} \Rightarrow \bar{\mathbf{D}}$). In this situation:

- If $\mathbf{c}_B^T \mathbf{B}^{-1} \bar{\mathbf{D}} - \mathbf{c}_D^T \leq \mathbf{0}^T$: the current basis \mathbf{B} is still optimal and the optimal objective value does not change.
- If $\mathbf{c}_B^T \mathbf{B}^{-1} \bar{\mathbf{D}} - \mathbf{c}_D^T \not\leq \mathbf{0}^T$: the current basis \mathbf{B} is not optimal. Use *primal* simplex method to find the new optimal solution.

Note: Due to x_j is not in the current basis \mathbf{B} , only the column associated with x_j will change, namely, $\mathbf{B}^{-1} \mathbf{A}_j \Rightarrow \mathbf{B}^{-1} \bar{\mathbf{A}}_j$ and the value at row 0 associated with x_j will become $\mathbf{c}_B^T \mathbf{B}^{-1} \bar{\mathbf{A}}_j - c_j$.





Change in the constraint matrix \mathbf{A} : $\mathbf{A}_j \Rightarrow \bar{\mathbf{A}}_j$

Case 2: \mathbf{A}_j is a basic column ($\mathbf{B} \Rightarrow \bar{\mathbf{B}}$). In this situation, all values in the simplex tableau are affected and:

- If $\mathbf{c}_B^T \bar{\mathbf{B}}^{-1} \mathbf{D} - \mathbf{c}_D^T \leq \mathbf{0}^T$ and $\bar{\mathbf{B}}^{-1} \mathbf{b} \geq \mathbf{0}$: the current basis \mathbf{B} is still optimal and the new optimal objective value is $\mathbf{c}_B^T \bar{\mathbf{B}}^{-1} \mathbf{b}$.
- If $\mathbf{c}_B^T \bar{\mathbf{B}}^{-1} \mathbf{D} - \mathbf{c}_D^T \leq \mathbf{0}^T$ and $\bar{\mathbf{B}}^{-1} \mathbf{b} \not\geq \mathbf{0}$: the current basis \mathbf{B} is not optimal and dual optimality condition (primal feasibility) is not satisfied. Use *dual* simplex method to find the new optimal solution.
- If $\mathbf{c}_B^T \bar{\mathbf{B}}^{-1} \mathbf{D} - \mathbf{c}_D^T \not\leq \mathbf{0}^T$ and $\bar{\mathbf{B}}^{-1} \mathbf{b} \geq \mathbf{0}$: the current basis \mathbf{B} is not optimal and primal optimality condition (dual feasibility) is not satisfied. Use *primal* simplex method to find the new optimal solution.
- If $\mathbf{c}_B^T \bar{\mathbf{B}}^{-1} \mathbf{D} - \mathbf{c}_D^T \not\leq \mathbf{0}^T$ and $\bar{\mathbf{B}}^{-1} \mathbf{b} \not\geq \mathbf{0}$: the current basis \mathbf{B} is not optimal and both primal & dual optimality conditions are not satisfied. Use *two-phase* or *Big M* method to find the new optimal solution.





Change in the constraint matrix \mathbf{A} : $\mathbf{A}_j \Rightarrow \bar{\mathbf{A}}_j$

Note: Due to $\mathbf{B} \Rightarrow \bar{\mathbf{B}}$, recalculations of $\mathbf{c}_B^T \bar{\mathbf{B}}^{-1} \mathbf{D} - \mathbf{c}_D^T$, $\bar{\mathbf{B}}^{-1} \mathbf{b}$ and other values of the simplex tableau becomes very complicated (it is not so easy to find $\bar{\mathbf{B}}^{-1}$!):

- Replace the column vector associated with x_j in the simplex tableau by $\mathbf{B}^{-1} \bar{\mathbf{A}}_j$ and $\mathbf{c}_B^T \mathbf{B}^{-1} \bar{\mathbf{A}}_j - c_j$ (note: still use \mathbf{B}).
- Pivot at y_{jj} to make x_j become a basic variable.



Change in the constraint matrix \mathbf{A} : $\mathbf{A}_j \Rightarrow \bar{\mathbf{A}}_j$

Example 4: Reconsider the problem in Example 2

- Replace $\mathbf{A}_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ by $\bar{\mathbf{A}}_2 = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$:

Since x_2 is a nonbasic variable, only the column associated with x_2 is affected.

$$\mathbf{B}^{-1}\bar{\mathbf{A}}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 5 \end{bmatrix} = \begin{bmatrix} 2 \\ 7 \end{bmatrix}, \quad \mathbf{c}_B^T \mathbf{B}^{-1} \bar{\mathbf{A}}_2 - c_2 = \begin{bmatrix} -2 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 7 \end{bmatrix} - 1 = -5$$





Change in the constraint matrix A : $A_j \Rightarrow \bar{A}_j$

The new simplex tableau:

	Z	x_1	x_2	x_3	x_4	x_5	RHS
Z	1	0	-5	-1	-2	0	-12
x_1	0	1	2	1	1	0	6
x_5	0	0	7	1	1	1	10

This is still the optimal one.





Change in the constraint matrix \mathbf{A} : $\mathbf{A}_j \Rightarrow \bar{\mathbf{A}}_j$

- Replace $\mathbf{A}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ by $\bar{\mathbf{A}}_1 = \begin{bmatrix} 3 \\ 6 \end{bmatrix}$:

Since x_1 is a basic variable, the whole simplex tableau will be affected. The new simplex tableau can be determined as follows:

$$\mathbf{B}^{-1}\bar{\mathbf{A}}_1 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 6 \end{bmatrix} = \begin{bmatrix} 3 \\ 9 \end{bmatrix}$$

$$\mathbf{c}_B^T \mathbf{B}^{-1} \bar{\mathbf{A}}_1 - c_1 = \begin{bmatrix} -2 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 9 \end{bmatrix} - (-2) = -6 - (-2) = -4$$



b ⇒ b'



Change in the constraint matrix A : $A_j \Rightarrow \bar{A}_j$

	Z	x_1	x_2	x_3	x_4	x_5	RHS
Z	1	-4	-3	-1	-2	0	-12
x_1	0	3*	1	1	1	0	6
x_5	0	9	3	1	1	1	10





Change in the constraint matrix A : $A_j \Rightarrow \bar{A}_j$

Pivot at 3^* , the new simplex tableau associated with the change from A_1 to \bar{A}_1 is:

	Z	x_1	x_2	x_3	x_4	x_5	RHS
Z	1	0	$-5/3$	$1/3$	$-2/3$	0	-4
x_1	0	1	$1/3$	$1/3$	$1/3$	0	2
x_5	0	0	0	-2	-2	1	-8

From the above simplex tableau, optimal solution can be found (use two-phase or Big M method)





Add a new variable: $x_k (c_k, \mathbf{A}_k)$

In this case, introduce a column associated with the newly added variable into the current simplex tableau. The elements of the new column can be determined as:

$$\mathbf{B}^{-1} \mathbf{A}_k \text{ and } \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{A}_k - c_k$$



Add a new variable: $x_k (c_k, \mathbf{A}_k)$

Example 5: Reconsider the problem in Example 2

Add x_6 with $c_6 = 1$ and $\mathbf{A}_6 = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$. We have:

$$\mathbf{B}^{-1}\mathbf{A}_6 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{A}_6 - c_6 = [-2 \quad 0] \begin{bmatrix} -1 \\ 1 \end{bmatrix} - 1 = 1$$



b ⇒ b'



Add a new variable: $x_k (c_k, A_k)$

The new simplex tableau:

	Z	x_1	x_2	x_3	x_4	x_5	x_6	RHS
Z	1	0	-3	-1	-2	0	1*	-12
x_1	0	1	1	1	1	0	-1	6
x_6	0	0	3	1	1	1	1	10

Not an optimal one: Use simplex method to find optimal solution.





Add a new constraint

Introduce a new row into the current simplex tableau and deal with the revised problem as if it is a new one. Add slack, surplus, or artificial variables if needed to form an initial basic solution

Example 6: Reconsider the problem in Example 2

Add a new constraint: $-x_1 + 2x_3 \geq 2$



Add a new constraint

Change the sign of the constraint and introduce the slack variable x_6 :

	Z	x_1	x_2	x_3	x_4	x_5	x_6	RHS
Z	1	0	-3	-1	-2	0	0	-12
x_1	0	1	1	1	1	0	0	6
x_5	0	0	3	1	1	1	0	10
x_6	0	1	0	-2	0	0	1	-2

Add a new constraint

Rearrange to form a simplex tableau as follows:

	Z	x_1	x_2	x_3	x_4	x_5	x_6	RHS
Z	1	0	-3	-1	-2	0	0	-12
x_1	0	1	1	1	1	0	0	6
x_5	0	0	3	1	1	1	0	10
x_6	0	0	-1	-3	-1	0	1	-8

Dual simplex method can be applied in this case.





Concluding Remarks

- When new information becomes available and needed to be incorporated into the optimization problem, the former optimal solution should be used as the **starting solution** to help find the updated optimal solution because the new optimal solution might be very closed to the former one.
- This will help to reduce the time needed to find the new optimal solution, and hence, make appropriate decisions in nearly real-time

