# Operations Research, Spring 2017
# Case Assignment 1

Ling-Chieh Kung*

IEDO Meat is a Taiwan-based big company running retail stores to sell meat. It sells nothing but meat because the CEO (chief executive officer) Luffy loves to eat meat. After years of successful operations in Taiwan, IEDO has accumulated great know-how in the retailing industry. Now it plans to go abroad to open retail stores in the southeast Asia. The CEO Luffy and vice president Zoro have chosen a particular county in a particular country to be the first trial of going abroad. The CMO (chief marketing officer) Nami has chosen 100 great places to build retail stores and made careful forecast of future demands. It was also determined that only one kind of meat will be sold there (at least during the beginning stage). The head of the manufacturing team Sanji has spent several months to make that meat popular by all consumers in that county. Everything looks fine.

Robin was hired into IEDO two weeks ago as the head of the Operations Research team. She was asked to deal with the logistic and replenishment problem faced by IEDO. The associate head of the OR team, Vivi, is experiences and knows the problem in details. However, she does not have helpful skills to solve the problem. Vivi explained all she knows to Robin as follows.

## 1 Problem description

In the county, the IEDO company has already determined to open 100 retail stores. The *expected daily demands* are provided in the spreadsheet “Store”, in which Column A contains store IDs and Column B contains expected daily demands. IEDO has determined that in every evening each store will be replenished exactly the amount of its expected daily demand. The locations of stores, which are expressed as $(x, y)$ coordinates measured in kilometers, are provided in Columns C and D in the spreadsheet.

*Department of Information Management, National Taiwan University. E-mail: lckung@ntu.edu.tw.

Now, IEDO wants to build several distribution centers (DCs) to do replenishment. According to the head of the construction team, Franky, there are ten candidate locations to build DCs. If a DC is going to be built in a location, IEDO also needs to determine its *scale*, i.e., the number of products that can be stored in it. Obviously, a larger DC will cost more. The variable cost per product of maintaining the operations in each DC (if it is built) is provided in the spreadsheet "DC". The second type of cost is the fixed amount of construction cost that one must pay to build a DC, regardless of its scale. The total cost of building a DC to a certain scale is thus the sum of the *construction cost*, which is independent of the scale, and the *maintenance cost*, which depends on the scale. In the spreadsheet "DC", column A contains DC IDs, column B contains the maintenance costs per unit product, and column C contains the fixed construction costs. The $(x, y)$ coordinates of the ten locations, measured in kilometers, are provided in Columns D and E. The associate head of the construction team Brook reminds Robin that the scales of DCs are not unlimited. The maximum scales of DCs are provided in Column F of the spreadsheet.

Once DCs are built, IEDO must determine how to replenish stores from DCs. The head of the retail operation team Chopper mentions that, ideally, a store should be replenished by only one DC to minimize the amount of time spent on receiving products. Such a way is called *single sourcing* in the operations management literature. However, as this restriction may increase the complexity of the planning, Robin decides to ignore this issue for a while and apply *multiple sourcing*. By doing so, a store may be replenished by one or multiple DCs so that the sum of all products it receives equals its expected daily demand. The *replenishment cost* for a DC to replenish a store is roughly proportional to the traveling distance times the replenishment amount. For ease of exposition, the unit cost of shipping one product to travel one kilometer has been normalized to \$1. In other words, if a DC travels for 10 kilometers to ship 10 products to a store, the replenishment cost is \$100.

The head of the shipment team Usopp makes further explanations about shipping. As the roads in the county are pretty much built in east-west and north-south directions, the traveling distance between two locations should be calculated as Manhattan distances, not Euclidean distances.

## 2 Two levels of decision making

Given all the information, Robin's mission is to develop a systematic way to determine three things: The locations to build DCs, scales of built DCs, and a replenishment plan to assign each store to a DC or multiple DCs. If a store is assigned to multiple DCs, how the DCs should split the replenishment work to fulfill the expected daily demand needs also be determined. Robin needs to make sure that all stores get the amounts of products they need while minimizing the total cost.

As a well trained operations researcher, Robin knows that the fixed construction costs introduce great difficulty to this problem. Therefore, she decides to divide the problem into two levels:[1]

1. In the upper level, she will determine the locations to build DCs. This level is called the *construction level*.
2. In the lower level, she will determine the scales of built DCs and the replenishment plan. This level is called the *SR level* (scale and replenishment level).

Robin clearly understand that, once a construction-level decision is made, the construction costs will become sunk costs that need not be considered in the SR level. The SR-level problem can then be solved by Linear Programming perfectly.

Even if the construction level problem cannot be directly solved by Linear Programming, being able to solve the SR level problem is still beneficial. For example, one may use intuition to propose several "good" ways to build DCs. Then each of these proposals may be evaluated by adding the construction costs and the operational costs of the *optimized* scale of replenishment plan.[2]

## 3 Tasks for this assignment

Each of the following task counts for 20 points.

1. Formulate a linear program whose solution can minimize the cost of the SR-level problem. Write a compact formulation. Clearly define your variables. Clearly explain your objective function and all your constraints.

[1] In the operations management literature, these levels are also called the *strategic* and *operational levels*, as the construction decision will last for many years, but the scale and replenishment may be adjusted more easily.

[2] In fact, if one is able to write a program (in C, C++, Java, etc.) to invoke a Linear Programming solver to solve the SR level problem, all the $2^{10} - 1$ ways of building DCs may be tested in one click. This method certainly fails if you have too many candidate locations, but it may work for just ten locations.

2. Use AMPL to write a model file and a data file that can solve the SR-level problem. Your model file should be a direct translation of your mathematical formulation made in Problem 1. It should contain no data; all numbers should be specified in the data file. Note that you need a given decision of DC construction. For this, use the solution contained in the spreadsheet "A Construction Plan".

3. Use your AMPL files in Problem 2 and the construction plan in the spreadsheet "A Construction Plan" to find an optimal scale and replenishment plan. Make a fine report (with texts, numbers, tables, figures, and concepts of information visualization) to present the plan.

4. For each of the ten locations to build DCs, determine whether it is worthwhile to expand its maximum scale. If no, explain why; if yes, quantify how worthy it is. Do the same thing to all expected daily demands and then suggest where to put more marketing budget to boost demands.

5. Propose a way to solve the construction level problem. Your method may not be perfect, and you do not need to implement your method. Just describe it and convince us that your method is effective (i.e., generating a solution whose total cost will be close to that of an optimal solution) and scalable (i.e., applicable when there are a lot of candidate locations).

## 4 Submission rules

Students should for teams to do this case assignment. A team should have ***at most four students***. A student cannot join two teams. The deadline of this homework is ***2 pm, March 27, 2017***. Please write down your answers, plans, suggestions, and interpretations into a report that is no longer than ten pages. Put a ***hard copy*** of your report into the instructor's mailbox on the first floor of Management Building 2. Moreover, put your report file, model file(s), data file(s), and whatever file(s) you have into ***a single ZIP file*** and submit to CEIBA. Works submitted between 2 pm and 3 pm will get 10 points deducted as a penalty. Submissions later than 3 pm will not be accepted. Only one hard copy and one zip file should be submitted by a team. Submitting more than one zip file or non-ZIP files will get 10 points off. Discussions among teams are welcome, but copying will results in severe penalties to everyone involved.

This case assignment counts for 10% for the final semester grades. Please do spend time and efforts on it!