Distribution Center Optimization using the Evolutionary Algorithm

Tellef Solberg

10/2/2020

1. Context

In industries delivering physical goods, it is of great importance to determine the optimal locations for distribution centers and thus minimize the total travelling distance to the demand units. If the company delivering the goods values short response times and incurs high costs on transportation the problem is of even greater importance. Using a mixed integer program with the aim of minimizing the total distance between the distribution centers and demand units, is a solution that has on many occations proven to be useful for analyzing such problems.

2. Problem

In this project we use x-y coordinates on a 2-dimensional plane, insert the demand units and locate the distribution centers in optimal positions. We have in total 25 demand units and 3 distribution centers.

The problem this project is aiming to solve is: *When we have 25 demand units and assigning 3 distribution centers to serve them, what are the locations that minimizes total travelling distance?*

3. Methodology

In order to minimize the total travelling distance for each distribution center, we construct a mixed integer program where we define the decision variables, parameters, constraints and an objective function. The solution of the program will be generated using the *evolutionary algorithm* built into the *Excel solver tool*.

The evolutionary algorithm is an algorithm with a wide range of applications in e.g. in planning, routing and classification. The algorithm relies on the fitness of each individual i and determines the optimal solution in a non-deterministic fashion (Moraglio, *Introduction to Natural Computation*). For the sake of project progression we refer to Alberto Moraglio's description of the algorithm provided in the reference list in the last chapter. For now, consider the following description of the algorithm:

1. Generate the initial population *P(0)*
2. Evaluate the fitness of each individual in *P(0)*
3. Generate offsprings using *variation operators* to form *P(i)*
4. Evaluate the fitness of each individual in *P(i)*
5. Select parents from *P(i)* and *P(i-1)* based on their fitness
6. Reiterate the algorithm until the halting criteria are satisfied.

Once the algorithm is applied to the mathematical program, we will obtain 3 different xy-coordinates representing the distribution centers, and they will be visualized in the 6th chapter.

4. Data Description

The input data to the model consists of 25 observations each being a demand unit with distinct x-y coordinates inidcating their different positions on the 2-dimensional plane. A summary of the data is given:

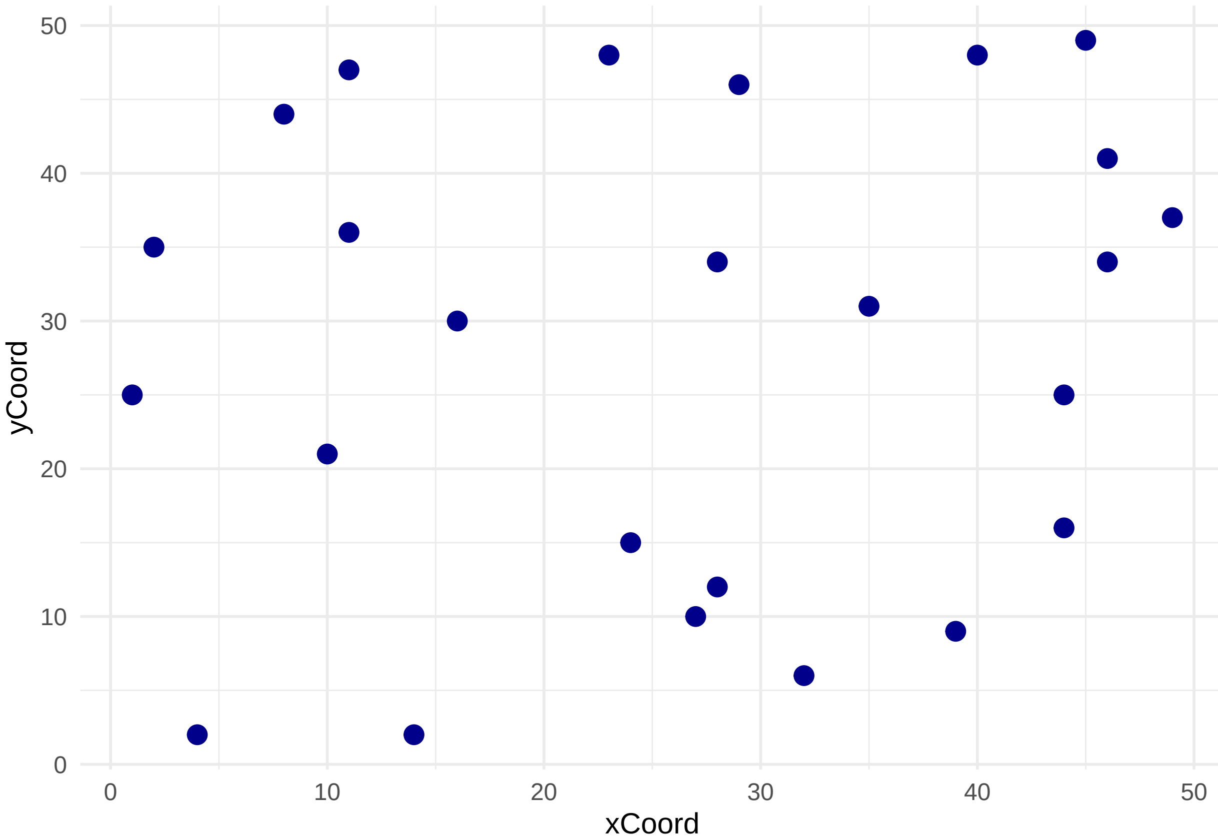
```
##       x Coord y Coord
##  [1,]      23      48
##  [2,]      24      15
##  [3,]       8      44
##  [4,]      27      10
##  [5,]       4       2
##  [6,]      28      12
##  [7,]      29      46
##  [8,]      49      37
##  [9,]      16      30
## [10,]      11      47
## [11,]      39       9
## [12,]      14       2
## [13,]      35      31
## [14,]      28      34
## [15,]      11      36
## [16,]      44      16
## [17,]      10      21
## [18,]       1      25
## [19,]       2      35
## [20,]      44      25
## [21,]      46      41
## [22,]      32       6
## [23,]      46      34
## [24,]      45      49
## [25,]      40      48
```

```
##     x Coord          y Coord
##  Min.   : 1.00   Min.   : 2.00
##  1st Qu.:11.00   1st Qu.:15.00
##  Median :28.00   Median :31.00
##  Mean   :26.24   Mean   :28.12
##  3rd Qu.:40.00   3rd Qu.:41.00
##  Max.   :49.00   Max.   :49.00
```

For a richer interpretation of the 25 demand units, we visualize them on a 2-dimensional plane using the R-package ggplot2:

```
require(tidyverse)
```

```
xyCoord1Plot <- ggplot(data = xyCoord1, aes(x=xCoord, y=yCoord)) +
  geom_point(color = "darkblue", size = 3) +
  theme_minimal()
print(xyCoord1Plot)
```



The blue scatters (hereafter refered to as nodes) presents each demand unit in the given problem, and three red nodes will be added after the algorithm has finished.

5. The Mathematical Model

In this section we formulate the mixed integer program with the corresponding objective function for minimization of travelling distance. First, we display parameters (data inputs), then we present the decision variables to be altered optimally and applied to the final objective function with the a set of given constraints that are realistic to the application.

Parameters

n = number of demand units

m = number of distribution centers

XYD_i = xy-coordinate for demand unit i

Decision variables:

A_{ij} = demand unit i assigned to the distribution center j

XYC_j = xy-coordinate for distribution center i

Objective function:

$$Min(\sum_{j=1}^{m} XYC_j - \sum_{i=1}^{n} XYD_i)$$

Subject to:

$50 \geq XYC_j > 0$

$A_{ij} \geq 1$

$A_{ij} \leq 3$

$A_{ij} = integer$

With the presented mixed integer program, we can run the evolutionary algorithm and dynamically optimize based on the chaning fitnesses of the individuals and the indivduals in relations to each other. Now, the the optimal mix of assigned demand units to distribution centers will be generated as we will inspect in the next chapter.

6. Interpretation and Analysis of the Results

The results obtained when running the mixed integer program yielded the following results

```
##    x  y
## 1 11 36
## 2 42 38
## 3 27 10
```

In the figure below, the optimal locations are determined by the red nodes which correspond to the xy-coordinates given in the tables below. Under the viualization, there is an added table showing which distribution center is assigned to which demand unit.

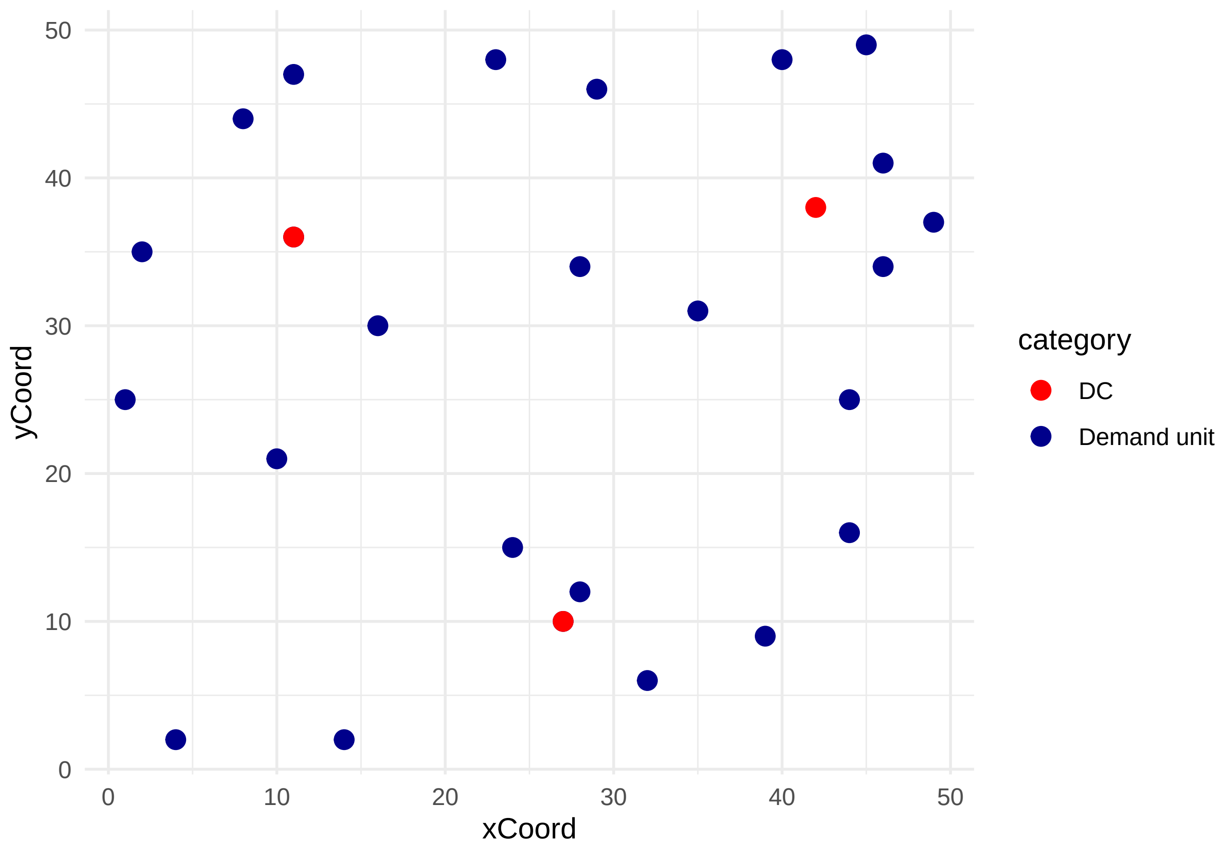
```
xyCoord1$cat <- "Demand unit"
xyCoord2$cat <- "DC"

colnames(xyCoord2) <- c("xCoord", "yCoord", "category")
colnames(xyCoord1) <- c("xCoord", "yCoord", "category")

xyCoord3 <- rbind(xyCoord1, xyCoord2)

xyCoord2Plot <- ggplot(data = xyCoord3, aes(x = xCoord, y = yCoord, color = category))+
  geom_point(size = 3) +
  theme_minimal()

xyCoord2Plot + scale_color_manual(values = c("red","darkblue"))
```



Thus, when optimizing the mixed integer program we obtain the three positions for the distribtuin centers: [11,36], [42,39] and [27,10]. It should be noted that the third distribution center with the values [27,10] is placed on top of the 4th observation in the initial dataset due to it only operating with integer values.

```
##       x Coord y Coord Assigned DC
##  [1,]      23      48           1
##  [2,]      24      15           3
##  [3,]       8      44           1
##  [4,]      27      10           3
##  [5,]       4       2           3
##  [6,]      28      12           3
##  [7,]      29      46           2
##  [8,]      49      37           2
##  [9,]      16      30           1
## [10,]      11      47           1
## [11,]      39       9           3
## [12,]      14       2           3
## [13,]      35      31           2
## [14,]      28      34           2
## [15,]      11      36           1
## [16,]      44      16           3
## [17,]      10      21           1
## [18,]       1      25           1
## [19,]       2      35           1
## [20,]      44      25           2
## [21,]      46      41           2
```

```
## [22,]       32       6             3
## [23,]       46      34             2
## [24,]       45      49             2
## [25,]       40      48             2
```

Final Remarks

In this project, we optimized the position of 3 distribution centers on a 2-dimensional plane using the evolutionary algorithm on the constructed mixed integer program. The distance minimizing locations yielded the xy-coordinates;[11,36], [42,38] and [27,10]. Note that since this problem is integer-based, the distribution center with he placement [27,10] is placed on top of the 4th observation in the initial dataset. If the algorithm was to operate on more exact xy-coordinates with an increased number of decimals, the analysis would provide more precise results.

Going forward, this sort of problem could be extended to e.g. the travelling salesman problem, using the Google Maps API for real life applications for designing optimal supply chains and analyzing whether this optimal soultion is feasible in relation to regulations/other natural constraints.

References

Alberto Moraglio *Examples and Design of Evolutionary Algorithms* from *Introduction to Natural Computation*

Curtis Frye *Using Solver for decision analysis* from *Lynda.com*

Disclaimer: This is not an academic or commerical project, it is only meant to be a way to wrap up my personal research on my repositories at www.Github.com/TellSol.