جامعة الحسين التقنية
Al Hussein Technical University

Course : **data structures**

Name of Student : **Mohammad Yahia Al-Qudah**

Student ID # : **18110016**

Student's email :**mohammad.alqudah@htu.edu.jo**

https://github.com/mohammad-alqudah/data-structures-

Contents

Introduction

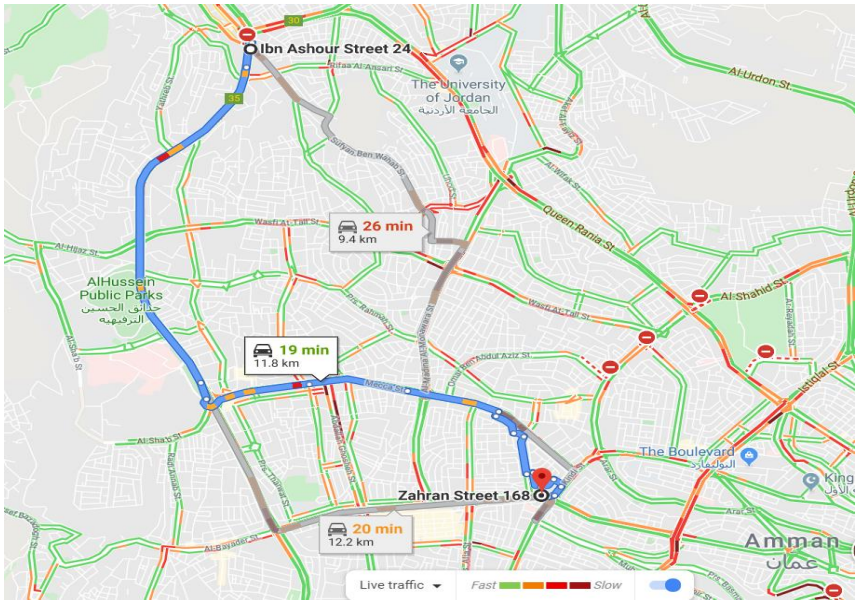
In this report, We will discuss an idea that will reduce traffic crisis in Amman, We will start from analyzing the problem first, then try to search for a mechanism to obtain the largest possible amount of data, Then try to use this data in solutions to Reducing the traffic crisis, After finding the solution, We will explain all parts of the solution, By explaining the solution design, and the algorithms used, with their advantages and disadvantages, We will also critically evaluate the solution at the end and make several recommendations for the solution

The problem

After tracking the traffic crisis in Amman, We have noticed that most of the time, the traffic crisis is concentrated in the part close to the University of Jordan Street, And Medina Street, While the surrounding streets are often free from the traffic crisis, Also, many streets face the same problem, and many drivers, they head to the crowded streets, And they leave empty streets because they don't know it is faster

This image taken by Google Maps shows, When moving from Sweileh (for example) to the Fifth Circle (for example), there are many options available that may vary in the time

In this example, there are many options that allow the driver to take one of them, It was also noted the time may increase to double sometimes



Also, cars entering Amman exceeds 177,000, according to traffic department statistics, and we are Knowing that the driver can enter Amman from more than one road

Solution

Through studying the problem, it became clear that the best solution is to design an intelligent program Able to guide drivers coming from outside Amman to the fastest way to their destination, where drivers can access this data through their smartphones or through the Informative boards that are on the way to Amman and The driver can also find the fastest route between any two points inside Amman,

This solution reduces the time of cars traveling in Amman, By reaching the best road that leads them to Amman

In addition, The program helps distribute cars to all available streets, For example, 11,000 cars cross the Jordan University Street every hour, if we distribute 3,000 cars on the surrounding streets, we can definitely reduce traffic congestion On University Street, which will reduce travel time

Input data

After searching for data that can help us develop the program, We found the following:

Available data for use:

1. Street distance
2. Number of lanes
3. the average length of the car
4. Safety distance between cars
5. Time to cross the street without a traffic crisis

Unavailable data for use:

1. Time to cross the street with a traffic crisis
2. Street threshold
3. The number of cars on each street
4. Accidental things that may delay traffic such as traffic lights, maintenance work

Some of the not available data can be found indirectly:

- We can find the street threshold by this equation

$$\frac{Street\ distance * Number\ of\ lanes}{length\ of\ the\ car + Safety\ distance}$$

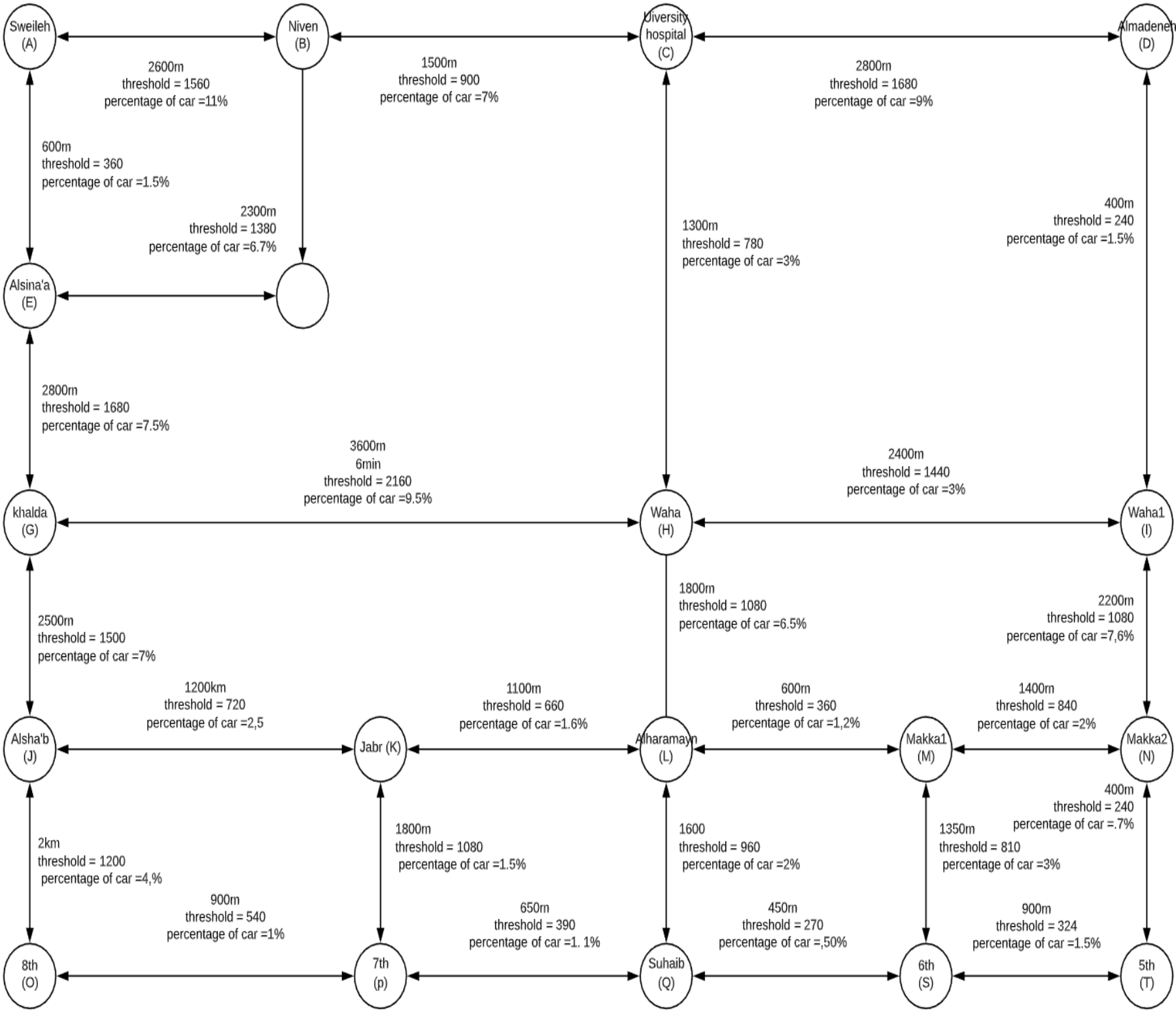
- We can find the numbers of cars in the streets through the Traffic Department in Amman, so that each street contains surveillance cameras, we can process the images in it and extract updated data all the time

  At the present time, we will use fake numbers by estimating the percentage of cars on each street and multiplying them by the number of cars in Amman (estimate)

  later on, we can develop software to fetch the data

This graph in the next page represents the available data that you will use in the program

جامعة الحسين التقنية
Al Hussein Technical University



The graph on the previous page was represented Threshold, distance street, The percentage of cars expected for most important streets in Amman, We have been ignoring the time because we cannot find it when there is crowding, But we know the higher the percentage of cars on the street, the inevitably increases the time

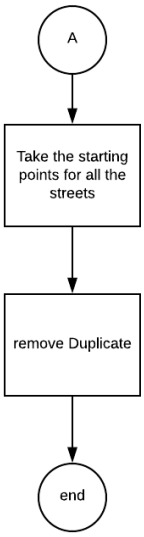
We will store static data on an excel page, the data should be includes Street start point, street endpoint, street distance, street threshold and percentage of cars expected in the street, Excel storage is chosen because it allows the user to easily add new data or modify existing data

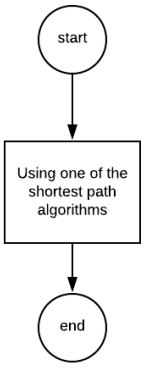
	A	B	C	D	E
1	from	to	distance	threshold	percentage of car
2	A	B	2600	1560	11
3	A	E	600	360	1.5
4	B	A	2600	1560	11
5	B	C	1500	900	7
6	C	B	1500	900	7
7	C	D	2800	1680	9
8	C	H	1300	780	3
9	D	C	2800	1680	9
10	D	I	400	240	1.5
11	E	A	600	360	1.5
12	E	G	2800	1680	7.5
13	E	B	2300	1380	6.6
14	G	E	2800	1680	7.5
15	G	J	2500	1500	7
16	G	H	3600	2160	9.5
17	H	G	3600	2160	9.5
18	H	C	1300	780	3
19	H	I	2400	1440	3
20	H	L	1800	1080	6.5
21	I	D	400	240	1.5
22	I	H	2400	1440	3
23	I	N	2200	1080	7.6
24	J	G	2500	1500	7
25	J	O	2000	1200	4
26	J	K	1200	720	2.5
27	K	J	1200	720	2.5
28	K	L	1100	660	1.6

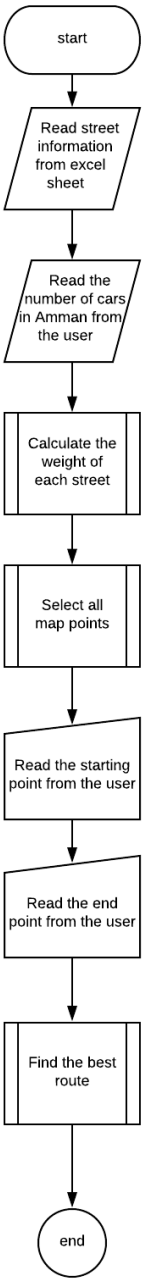
Design the solution

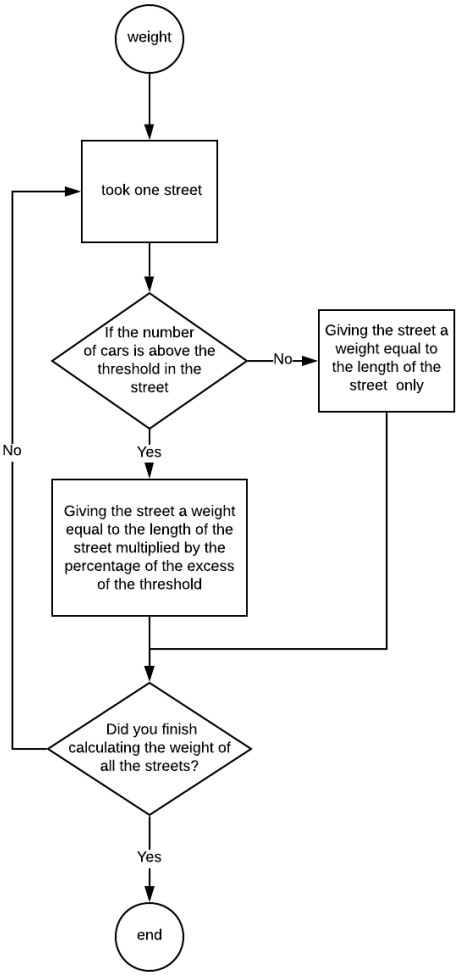
After analyzing the available data , We will design the solution before writing the code , By drawing flowchart , Shows an outline of the program's steps , So that it clarifies the steps the program works from starting data entry and processing and exit steps , Design is very important, because the cost of modification to the design is much lower than the cost of modifying the solution

When drawing the flowchart we drew the basic operations of the program, then we analyzed each process into a separate flowchart , On the next page of flowchart where we will explain each part with a separate point


A
Take the starting points for all the streets
remove Duplicate
end


start
Using one of the shortest path algorithms
end


start
Read street information from excel sheet
Read the number of cars in Amman from the user
Calculate the weight of each street
Select all map points
Read the starting point from the user
Read the end point from the user
Find the best route
end


weight
took one street
If the number of cars is above the threshold in the street
No
Giving the street a weight equal to the length of the street only
Yes
Giving the street a weight equal to the length of the street multiplied by the percentage of the excess of the threshold
Did you finish calculating the weight of all the streets?
No
Yes
end

Then we designed a flowchart to illustrate the steps of the progra

1. The first step of the program that has been implemented is reading the data from the excel sheet

```
def readFromExcel():


    loc = (r"C:\Users\mohammad alqudah\PycharmProjects\untitled7\Book1.xlsx")
    Arr=[]
    wb = xlrd.open_workbook(loc)
    sheet = wb.sheet_by_index(0)
    sheet.cell_value(0, 0)

    for i in range(sheet.nrows):
       Arr.append([])
       Arr[i].append(sheet.cell_value(i, 0))
       Arr[i].append(sheet.cell_value(i, 1))
       Arr[i].append(sheet.cell_value(i, 2))
       Arr[i].append(sheet.cell_value(i, 3))
       Arr[i].append(sheet.cell_value(i, 4))
    Arr.remove(Arr[0])

    return (Arr)
```

We have used a list because it can contain more than one data type at the same time, Plus, its size is flexible, and we don't need to define a size at the beginning, the list can also be indexed

2. The second step is to read the number of cars from the user

```
def carsNumber():
    while True:
        try:
            while True:
                carsNumber = int(input("Enter Number of cars in Amman : "))
                if carsNumber > 0:
                    return carsNumber
                else:
                    print("The number of cars cannot be less than zero")
            break
        except:
            print("This is a string")
```

We designed this part of the code can validate the entries and request new entries if the input is incorrect Because the user is possible to enter incorrect values

3. In this step, we will calculate the cost of crossing each street

```
def Weght(Matrix,numberOfCar):

    weght=[]
    for row in Matrix:
        percentage =float(row[4]/100)
        threshold=(int(row[3]))
        distance=int(row[2])

        if  numberOfCar*percentage > threshold:
            Weght=(numberOfCar*percentage-threshold)/threshold*4*distance+distance
            weght.append([row[0],row[1],Weght])
        else: weght.append([row[0],row[1],distance])

    return weght
```

We used the list because it has great flexibility in storing data

4. Here we selected vertex and stored it in a list

```
def removeDuplicate(duplicate):
    final_list = []
    for row in duplicate:
        if row[0] not in final_list:
            final_list.append(row[0])
    return final_list
```

We selected vertex from the first column in Excel that contains the starting point of each street, then we removed the duplicate to get a list containing vertex without duplicate

5. In this step of the code, The user selects the starting or ending points

```
def selectPoint(vertexList):
    while True:
        Point = str(input("Enter starting point : "))
        Point = findInVertexList(vertexList, Point)
        if Point >= 0:
            return Point
        else:
            print("The entered letter must be from one of the following
letter",vertexList)
```

the user selects the starting or ending point, Then the code calls a function to convert the letter to a number ,

6.

```
def findInVertexList(list,Point):

    for i in range(len(list)):
        if str(list[i]) == Point:
            return i

    return -1
```

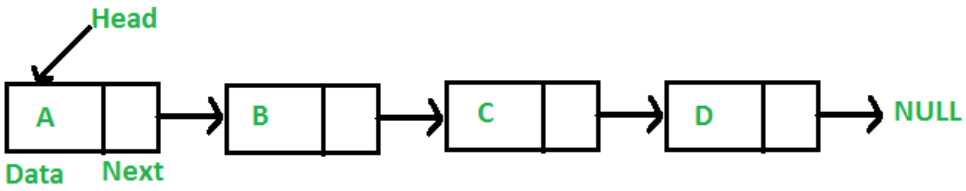
Data structures :

Are a specific way to organize and process data. There are many types of data, basic and advanced, these types are used depending on the purpose of the program. Data structures allow us to store information on the Harddisk,and allows us to manage big data easily and efficiently, increasing the speed and performance of programs ,

Therefore, the program was built on data structures to increase the efficiency of the program, because we need to process a huge amount of data at a high speed , Each type has specific uses So we studied the types, at first, to determine the best data structures we need

Linked List Data Structure:

They are linear data lists, in which data are stored as connected objects, not in a specific location in memory as in the traditional mode



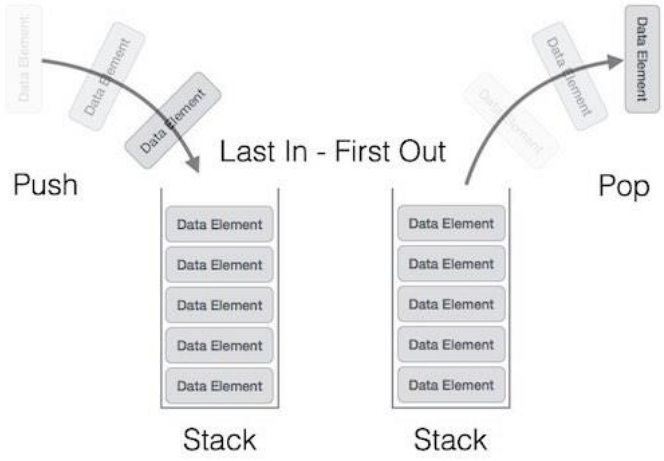
The data in the linked list is similar to the nodes where each node contains a set of data, and each node is linked to the node immediately following it, if we want to reach a node in the middle we will need to start from the beginning of the nodes and then walk on each node until we reach the required node

This type of data structure is very much applied

- In a web browser, where you go to the next page and the previous page via linked list
- In the image browser, we are allowed to go to the next and previous image via linked list
- How much the same idea is used in the music player and PowerPoint

Stack Data Structure:

Stack is a linear data structure which follows a particular order in which the operations are performed. The order may be LIFO(Last In First Out) or FILO(First In Last Out). (GeeksforGeeks, 2020)



This method is similar to the way dishes are arranged in real life, where if we have a set of dishes placed on each other, The process is the last dish that was placed is the first dish that will be taken

The basic operations of the stack

- `push()  Add an item to the stack`
- `pop() Remove an item from the stack`
- Peek or Top: Returns the top focus element.
- isEmpty: Returns true if the stack is empty, else false

`push()` To add an element to a stack, the computer performs several actions

1. Initially the device checks to see if the stack is full or not,
2. If it is full, the device generates an error message and stops
3. If it is not full, it moves to the next empty place
4. Then it adds the element to the empty spot,
5. then returns successfully

`pop()`

1. Initially the device checks to see if the stack is empty or not,
2. If it is empty, the device generates an error message and stops
3. If it is not empty, it removes an element
4. then returns successfully

GeeksforGeeks. (2020). *Stack Data Structure - GeeksforGeeks*. [online] Available at: https://www.geeksforgeeks.org/stack-data-structure/ [Accessed 26 Jan. 2020].