Business Intelligence Workplace

Report P03: Cola War

Student : Yassine Ammour
Instructor : Dr. Piotr Nyczka
Class : Business Intelligence Workplace

Introduction:

The Aim of this Project:

- The Aim of this Project is to build an agent based modeling simulation, based on modified version of sznajd model
- In a matrix we are gonna have too agents represent two opinions
- We are gonna have an initial size droplet with opinion (A)
- The rest is opinion (B)
- We are going to select a random agent
- and based on probability (p) we are gonna apply either Advertisement or conformity
- **Conformity** : check the selected agent's neighbors (Left, Right, Up, Down)
- If on of them is same opinion, make all neighbors the same
- **Advertisement** : Switch the selected agent to the opinion of the droplet
- The loop is gonna end when we reach a threshold (75 % of agent convinced)
- As inputs we are going to have a list of **Droplet sizes** and **probability values,** to apply different combinations
- At the end calculate the price based of a given function
- Repeat the simulation 10 times
- Average the results and make a bar plot to minimize the price and make a conclusion

Initialization:

- L*L Size doesn't change
- Empty matrices to save (Times, Droplet sizes, Ads cost, Prices)

```
L = 20;     %system size L x L
Time = []
Droplet = []
Ads = []
Price = []
```

Initialization:

- The mother loop is for repeating the operation 10 time (10 trials)
- List of our D (droplet size) and P (probability) for our combinations
- Initializing (cm) to count combinations

- Second and third loop for our combination (for every element of D with every element of P) do this :

1. Take one value of p and d
2. Set our matrix (L*L) to (-1)
3. Initialize the droplet and set it's elements to (+1)
4. Set a threshold to count number of ones we have

```
for trial = 1:1:10
    D = [ 0, 6, 10, 14]
    P = [0.08, 0.09, 0.10, 0.11, 0.12, 0.13, 0.14, 0.15]
    cm = 0
    for id= 1:1:size(D,2)
        for ip= 1:1:size(P,2)
            % Dock figures in the
            % main window
            set(0,'DefaultFigureWindowStyle','docked')
            % Initiale Parameters:
            p = P(ip);                          % Probability p (Advertising cost)
            d = D(id);                          % droplet size d x d
            T = 0;                              % initialize the Time
            S = zeros(L,L) - 1;                 % opinion matrix, initially set to -1
            S((L-d)/2+1:((L-d)/2)+d ,(L-d)/2+1:(L-d)/2+d)= 1;     % droplet, initially set to +1
            % Time loop
            tresh = sum(S(:) == 1)
```

- A While loop for Advertisement or conformity
- While we didn't reach our threshold
- With probability (p) Apply Advertisement :

1. Pick a random coordinates for an agent
2. Set him to (1)

1	x	1	=	1
-1	x	-1	=	1

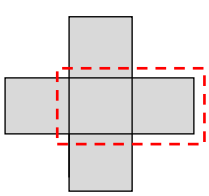
- Else : Conformity

1. Pick a random agent

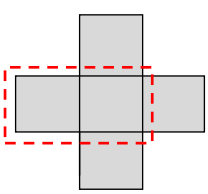
```
while (tresh <=round(L*L*75/100))
    T = T + 1
    tresh = sum(S(:) == 1)
    if rand() <=p
        % Apply Advertising
        i = floor(rand*L)+1;
        j = floor(rand*L)+1;
        S(i,j)=S(i,j).^2
    else
        %Conformity
        % Randomly select a cell (an agent), say S(i,j) with i,j in {1,…,N}
        i = floor(rand*L)+1;
        j = floor(rand*L)+1;
```

2. Check Conformity

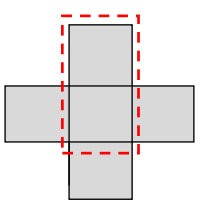
- We will check the conformity in all direction using a plus shape called checking area
- The center of the plus is the piked random agent
- To apply conformity it will check all direction, if similar agents found for one of them, it will apply conformity
- it will appear as a moving grey plus in the live matrix when running the code



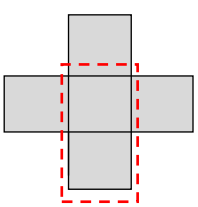
Check Right



Check Left

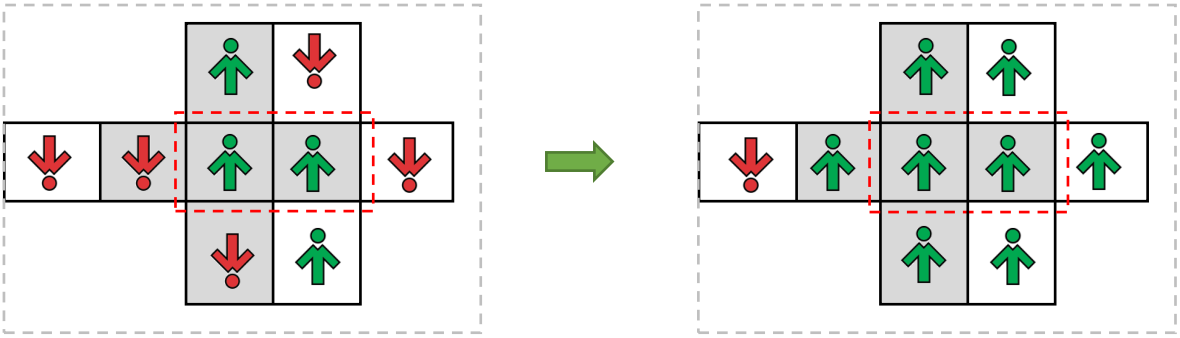


Check Up



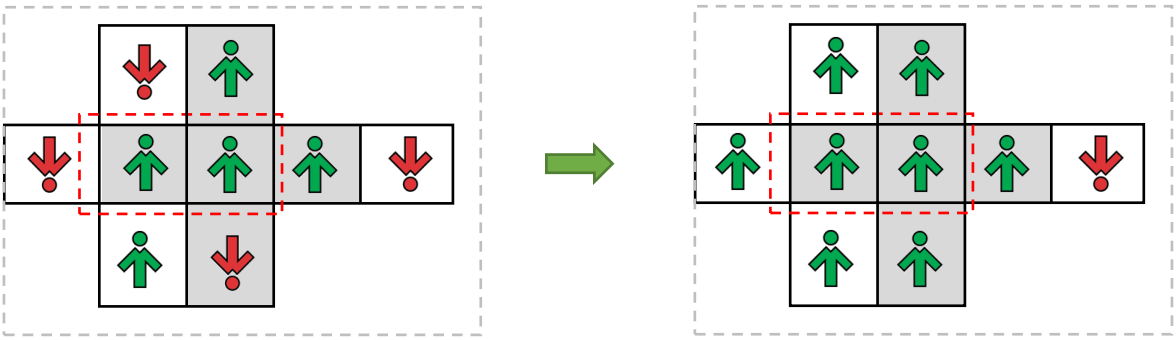
Check Down

1. Cheek Neighbor on the Right



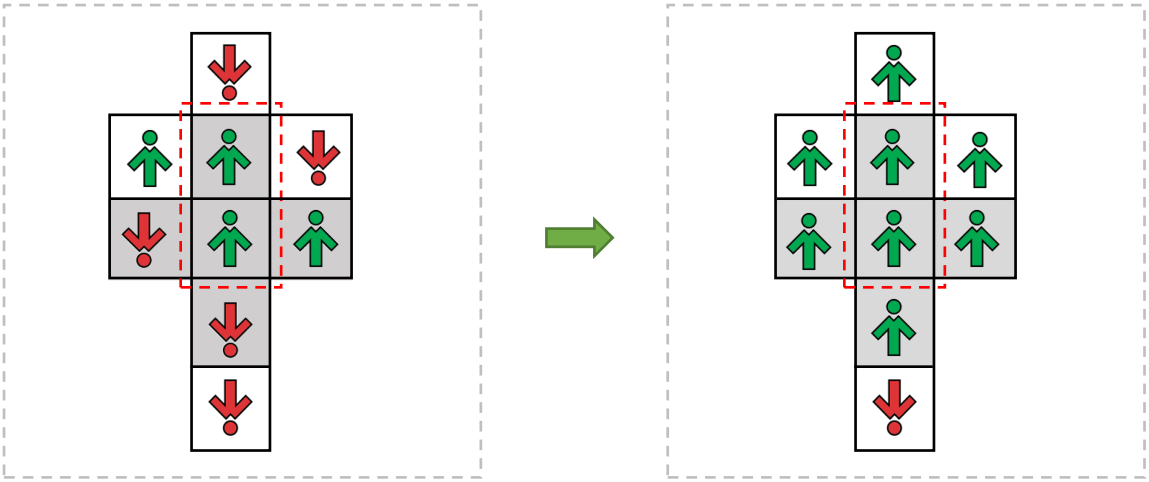
```
% Check if agents have the same opinion
% Check on the Right
if S(i,j)*S(i,mod(j+1-1,L)+1)==1
    % Modify agents above the panel
    S(mod(i-1-1,L)+1,j)=S(i,j);
    S(mod(i-1-1,L)+1,mod(j+1-1,L)+1)=S(i,j);
    % Modify agent to the left of the panel
    S(i,mod(j-1-1,L)+1)=S(i,j);
    % Modify agent to the right of the panel
    S(i,mod(j+2-1,L)+1)=S(i,j);
    % Modify agents below the panel
    S(mod(i+1-1,L)+1,j)=S(i,j);
    S(mod(i+1-1,L)+1,mod(j+1-1,L)+1)=S(i,j);
end
```

2. Cheek Neighbor on the left



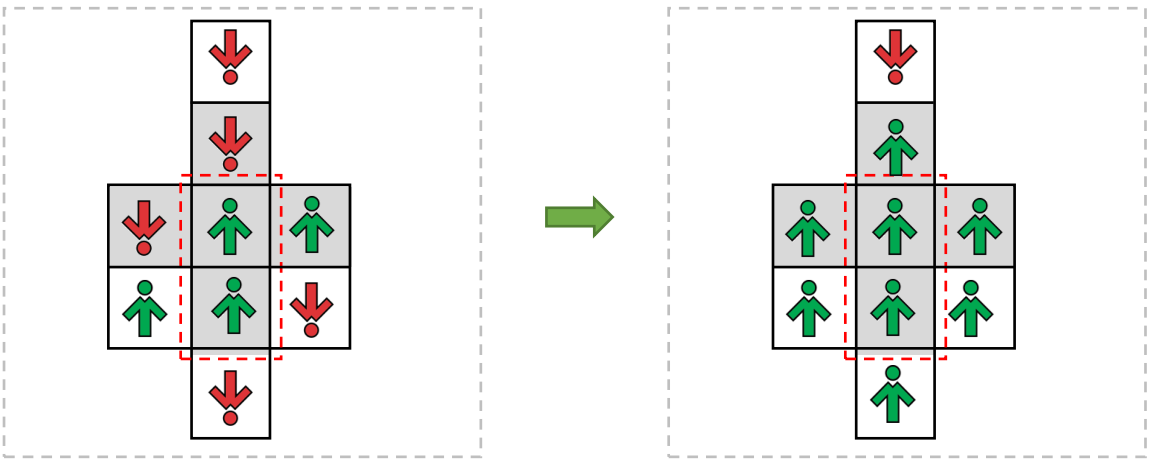
```
% Check on the Left
if   S(i,j)*S(i,mod(j-1-1,L)+1)==1
    % Modify agents above the panel
    S(mod(i-1-1,L)+1,j)=S(i,j);
    S(mod(i-1-1,L)+1,mod(j-1-1,L)+1)=S(i,j);
    % Modify agent to the left of the panel
    S(i,mod(j-2-1,L)+1)=S(i,j);
    % Modify agent to the right of the panel
    S(i,mod(j+1-1,L)+1)=S(i,j);
    % Modify agents below the panel
    S(mod(i+1-1,L)+1,j)=S(i,j);
    S(mod(i+1-1,L)+1,mod(j-1-1,L)+1)=S(i,j);
end
```

3. Cheek Up



```
% Check Up
if  S(i,j)*S(mod(i-1-1,L)+1,j)==1
    % Modify agents above the panel
    S(mod(i-2-1,L)+1,j)=S(i,j);
    % Modify agent to the right of the panel
    S(i,mod(j+1-1,L)+1)=S(i,j);
    S(mod(i-1-1,L)+1,mod(j+1-1,L)+1)=S(i,j);
    % Modify agent to the left of the panel
    S(i,mod(j-1-1,L)+1)=S(i,j);
    S(mod(i-1-1,L)+1,mod(j-1-1,L)+1)=S(i,j);
    % Modify agents below the panel
    S(mod(i+1-1,L)+1,j)=S(i,j);
end
```

4. Cheek Down



```
% Check Down
if  S(i,j)*S(mod(i+1-1,L)+1,j)==1
    % Modify agents above the panel
    S(mod(i-1-1,L)+1,j)=S(i,j);
    % Modify agent to the left of the panel
    S(i,mod(j-1-1,L)+1)=S(i,j);
    S(mod(i+1-1,L)+1,mod(j-1-1,L)+1)=S(i,j);
    % Modify agent to the right of the panel
    S(i,mod(j+1-1,L)+1)=S(i,j);
    S(mod(i+1-1,L)+1,mod(j+1-1,L)+1)=S(i,j);
    % Modify agents below the panel
    S(mod(i+2-1,L)+1,j)=S(i,j);
end
```

- This Is to plot the live matrix
- White agent = 1
- Black agent = -1
- Grey Plus Checking Area

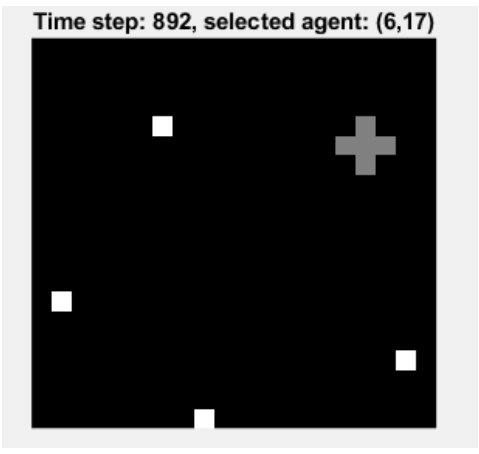
```
    %figure(1)
    % Plot selected panel in gray
    S1_old = S(i,j); S2_old = S(i,mod(j+1-1,L)+1);  S3_old = S(mod(i-1-1,L)+1,j);
    S4_old = S(mod(i+1-1,L)+1,j) ; S5_old = S(i,mod(j-1-1,L)+1)
    S(i,j) = 0; S(i,mod(j+1-1,L)+1) = 0; S(mod(i-1-1,L)+1,j)= 0; S(mod(i+1-1,L)+1,j)= 0
    S(i,mod(j-1-1,L)+1)= 0
    % Plot system state: 0 - black, 1 - white, (0,1) - gray
    imshow((S+1)/2,'InitialMagnification','fit')
        title(['Time step: '  num2str(T) ...
        ', selected agent: (' num2str(i) ',' num2str(j) ')'])
    pause(1)
    % Set panel state to original values
    S(i,j) = S1_old; S(i,mod(j+1-1,L)+1) = S2_old;
    S(mod(i-1-1,L)+1,j) = S3_old;  S(mod(i+1-1,L)+1,j) = S4_old;
    S(i,mod(j-1-1,L)+1) = S5_old
end
```

- At the end of the loop :

1. Increment the combination counter (cm)
2. Calculate the Price (X) using the given function
3. Finally save the (Time, Droplet size, Advertisement cost and the Price)

```
            end
            cm = cm + 1
            fp = 600*p.^6
            gd = d.^2
            X = fp*T + gd
            Time(trial,cm) = T
            Droplet(trial,cm)= d
            Ads(trial,cm)= p
            Price(trial,cm)  = X
        end
    end
end
```

- In the live plot it woold will appear like this



- White agent = 1
- Black agent = -1
- Grey area : checking area

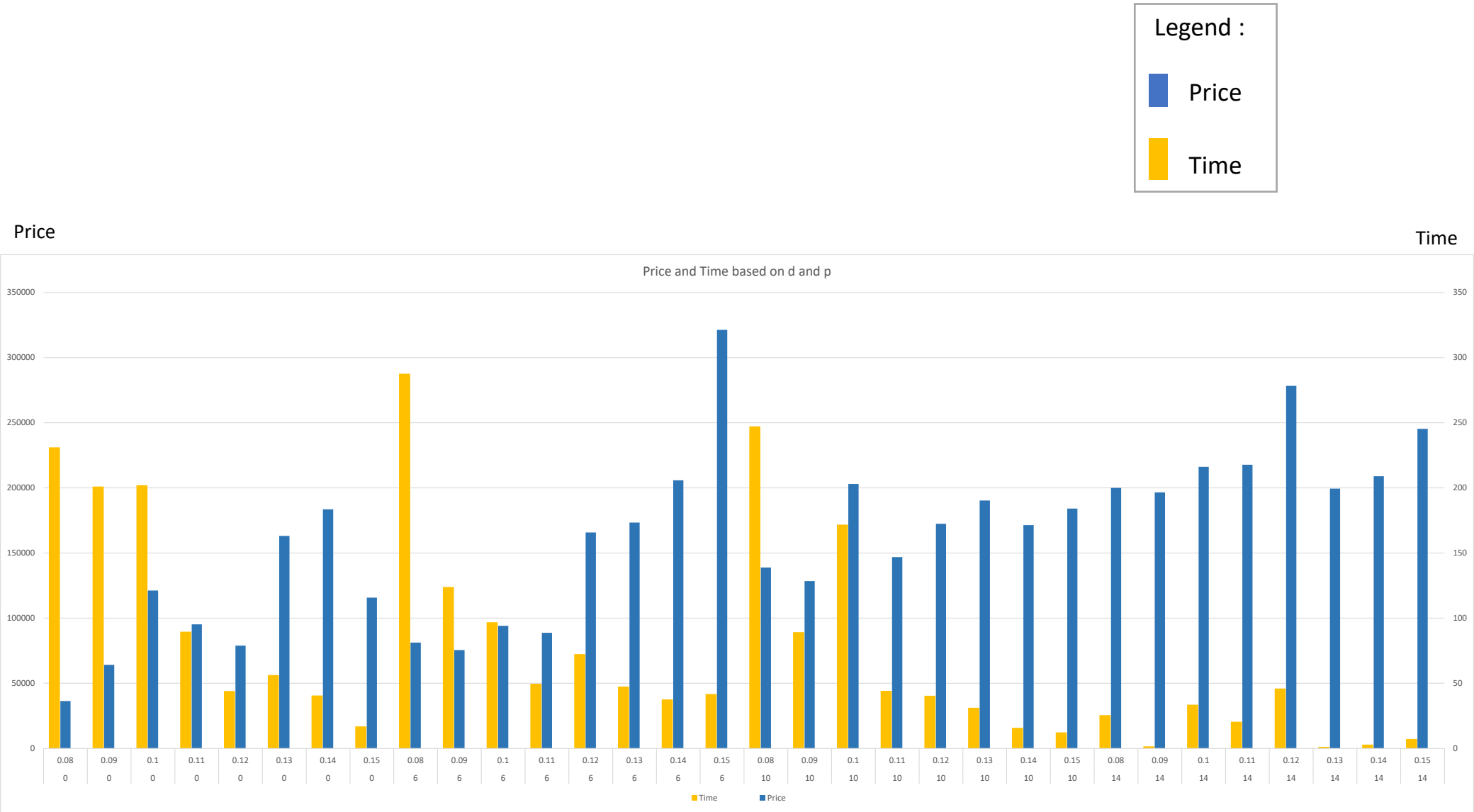
Bar Plot :

1. Exported the results from MATLAB
2. Average The results of 10 trials

Name	Value
Ads	*10x32 double*
cm	32
d	14
D	[0,6,10,14]
Droplet	*10x32 double*
fp	0.0068
gd	196
i	20
id	4
ip	8
j	18
L	20
p	0.1500
P	[0.0800,0.0900,0.1000,...
Price	*10x32 double*
S	*20x20 double*
T	4592
Time	*10x32 double*
tresh	303
trial	10
X	227.3834

	A	B	C	D
1	Droplet	Ads	Time	Price
2	0	0.08	231145.1	36.35576
3	0	0.09	201053.8	64.10885
4	0	0.1	202054.5	121.2328
5	0	0.11	89562.4	95.1994
6	0	0.12	44030	78.8829
7	0	0.13	56319.3	163.105
8	0	0.14	40621.6	183.518
9	0	0.15	16935	115.7399
10	6	0.08	287598.7	81.2358
11	6	0.09	123919	75.5139
12	6	0.1	96877.8	94.1272
13	6	0.11	49592.1	88.7134
14	6	0.12	72414.2	165.7362
15	6	0.13	47443.1	173.3966
16	6	0.14	37601.8	205.8738
17	6	0.15	41748.2	321.321
18	10	0.08	247226	138.885
19	10	0.09	89190.7	128.44
20	10	0.1	171738	203.043
21	10	0.11	44127.8	146.904
22	10	0.12	40453.2	172.473
23	10	0.13	31194.9	190.345

1. Plot the results (Next page)



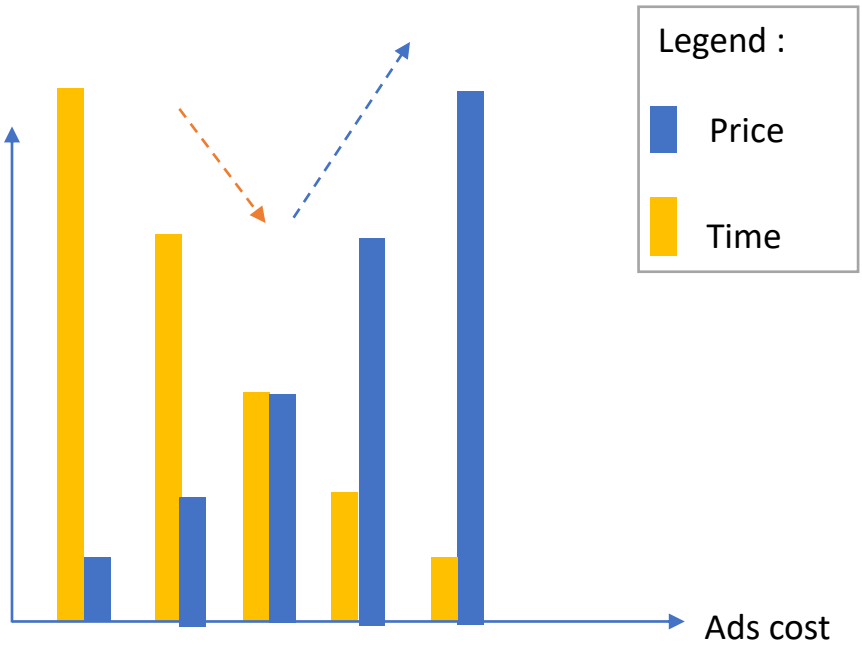
Bar Plot : Price and Time based on combinations of d and p

Conclusion

- For each individual droplet size, the Graph can be translated into this :

1. The higher the Ads cost, the more expensive our campaign would be
2. The more expensive our campaign, the less time we need to convince our audience

- And the bigger the droplet size the bigger the gap between time and price, as we notice that the bigger the droplet size, the more price is going up



Minimizing the price

- If we had to minimize the price we would choose the smallest values of (p) and (d) , first ever combination (d =0, p=0.08), but we are going to sacrifice the time
- So it depend on the advertiser, if he want **fast results**, he would **pay more**
- If he could afford waiting too long, he can pay the least amount of money
- The advertisement industry is a biding game, you pay more, you get what you want faster, so the advertisers always look for middle ground between not paying too much and not waiting too much