ABC INC. - RECOMMENDATIONS FOR MARKETING BUDGET ALLOCATION

INTRODUCTION

This report presents results from optimal allocation of marketing budget across platforms, and recommendations on the amount to be allocated to each platform, based on a given set of constraints. The allocation was treated as an optimization problem, solved using Python programming language. The problem statement began with allocating a budget of $10 million across 10 different marketing platforms, subject to the following constraints:

- The amount invested in Print and TV should be no more than the amount spent on Facebook and Email
- The total amount used in social media (Facebook, LinkedIn, Instagram, Snapchat, and Twitter) should be at least twice of SEO plus AdWords
- For each platform, the amount invested should be no more than $3 million

Later sections in the report detail about other scenarios and the solutions.

SOLUTION

The problem was solved using the Simplex algorithm. The Simplex algorithm is a mathematical optimization algorithm that can be used to solve linear programming problems. The Simplex algorithm works by iteratively improving a solution until it reaches an optimal solution.

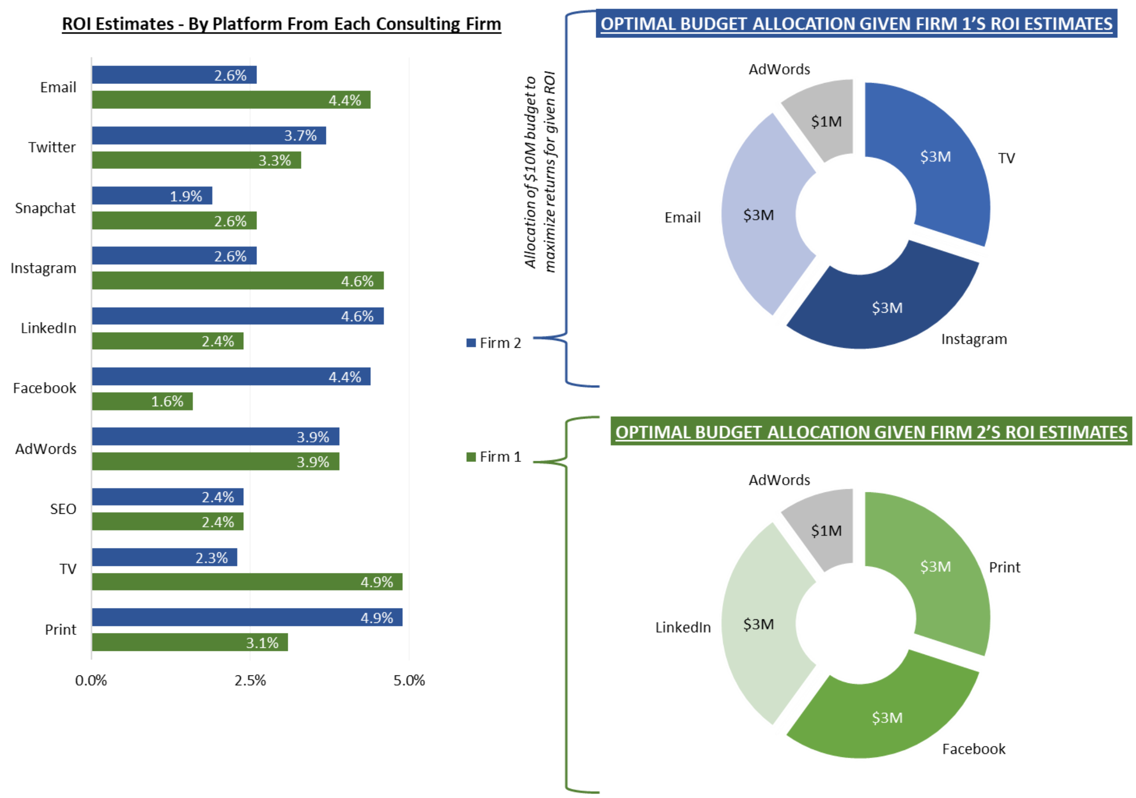
The 10 platforms available are: Print, TV, SEO, AdWords, Facebook, LinkedIn, Instagram, Snapchat
, Twitter, Email.

Comparison of Allocations - Using Different ROI Estimates

Return on investment (ROI) for each platform was estimated by an external consulting firm (referred to as ‘Firm 1’ henceforth). An allocation was computed using the ROI rates (referred to as ‘ROI 1’ henceforth) shared by Firm 1 to maximize returns. To err on the side of caution, the results were further validated by hiring a second consulting firm (referred to as ‘Firm 2’ henceforth) which provided a second set of ROIs (referred to as ‘ROI 2’ henceforth).

The results obtained for optimal allocation are below. Allocating the $10M budget as below would lead to the optimal (maximum) total returns:

- Firm 1-ROI 1: $3M each in TV, Instagram, Email & $1M in AdWords; giving an optimal return of $0.456M
- Firm 2-ROI 2: $3M each in Print, Facebook, LinkedIn & $1M in AdWords; giving an optimal return of $0.456M
- The allocations are not the same. The platforms to be invested in shift from TV, Instagram and Email to Print, Facebook and LinkedIn while AdWords pops up in both cases



Analysis of Optimal Returns - Effect of Optimal Allocations and Estimated ROI

- We observe that despite the same optimal return, the allocations are not the same for the 2 estimated sets of ROIs
- Taking the combination of estimated ROIs and the optimal allocation (derived from each estimated ROI), we compute the returns and estimate the deviation from optimal returns (which was $0.456M in either case)
- The results can be seen in the table below
    - In the first case, using ROI 1 and optimal allocation from ROI 2, reduces the returns by $0.204M
    - In the second case, using ROI 2 and optimal allocation from ROI 1, reduces the returns by $0.192M

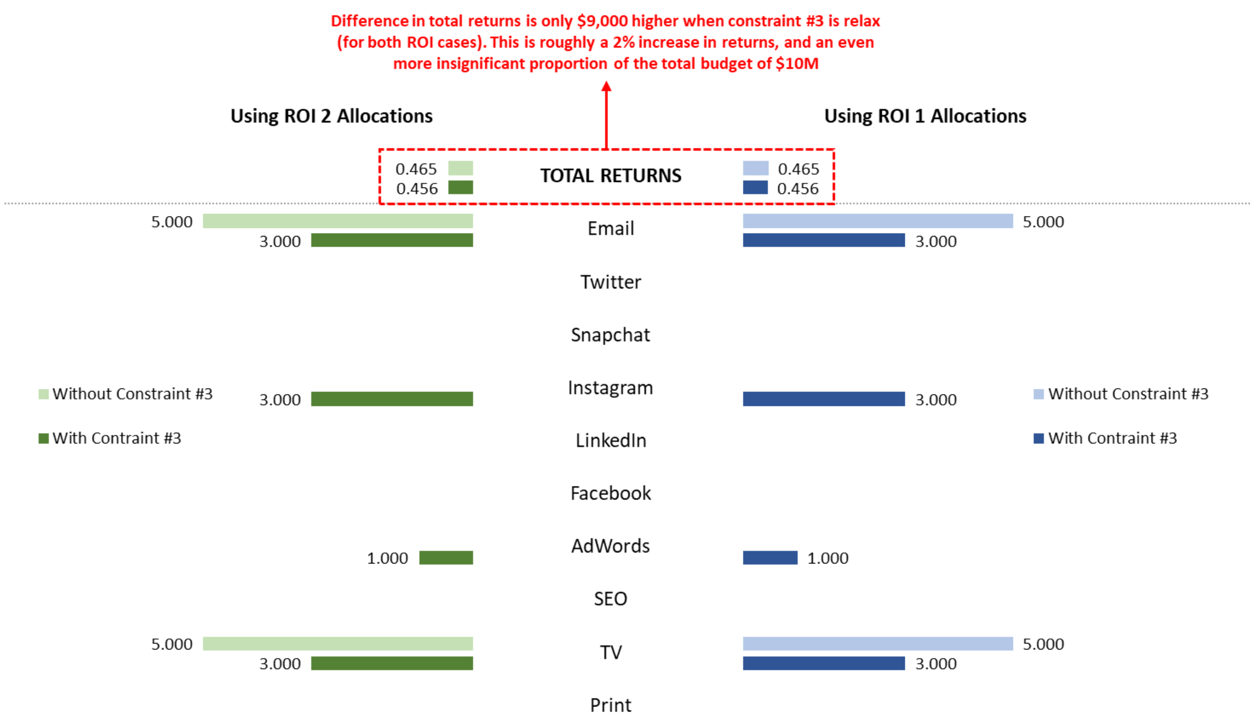
TOTAL RETURNS WITH VARYING ROI & ALLOCATION

TOTAL RETURNS (in Millions of Dollars)	USING FIRM 1 ROI	USING FIRM 2 ROI
USING OPTIMAL ALLOCATION FOR MAXIMUM RETURNS FROM FIRM 1 ROI	$0.456M	$0.264M
USING OPTIMAL ALLOCATION FOR MAXIMUM RETURNS FROM FIRM 2 ROI	$0.252M	$0.456M
DIFFERENCE	$0.204M	$0.192M

(Marginal) Effect of Constraint #3 (for each platform, amount invested should be no more than $3M)

To test the effect of constraint #3, the optimization was performed with and without the constraint, and then the optimal returns were calculated. This was performed taking both ROI 1 and ROI 2.

RETURNS AND ALLOCATION - WITH / WITHOUT CONSTRAINT #3



From the chart above, we see the allocation, while **still** being subject to constraints #1 and #2, does change and move to the high ROI platforms, in both cases.

However, the total returns for both cases go down **only by $9K** (which is insignificant given the total budget). This is a small price to pay, given that the constraint #3 ensures more diversity of platforms, which seems to be a practical decision, based on prior experiences.

Sensitivity Analysis - Obtaining Optimal Return with Changing ROIs

For this analysis, we take ROI 1 into consideration. The model is tweaked to test the limits on how much each platform's ROI can go up (and down), while still resulting in optimal return (of $0.456M). The least the ROIs can go to are termed as 'Lower bounds' and the most as 'Upper bounds.'

The table below lists the range of each platform's ROI which would still result in optimal returns. Sensitivity is a measure of this 'how much' change of each platform still allows the model to return the optimal solution. The infinities (both positive and negative) here denote that those respective ROIs can either go down to 0% or 100% (reading the infinities from a practical perspective) since they no longer keep the optimal solution within bounds.

SENSITIVITY ANALYSIS

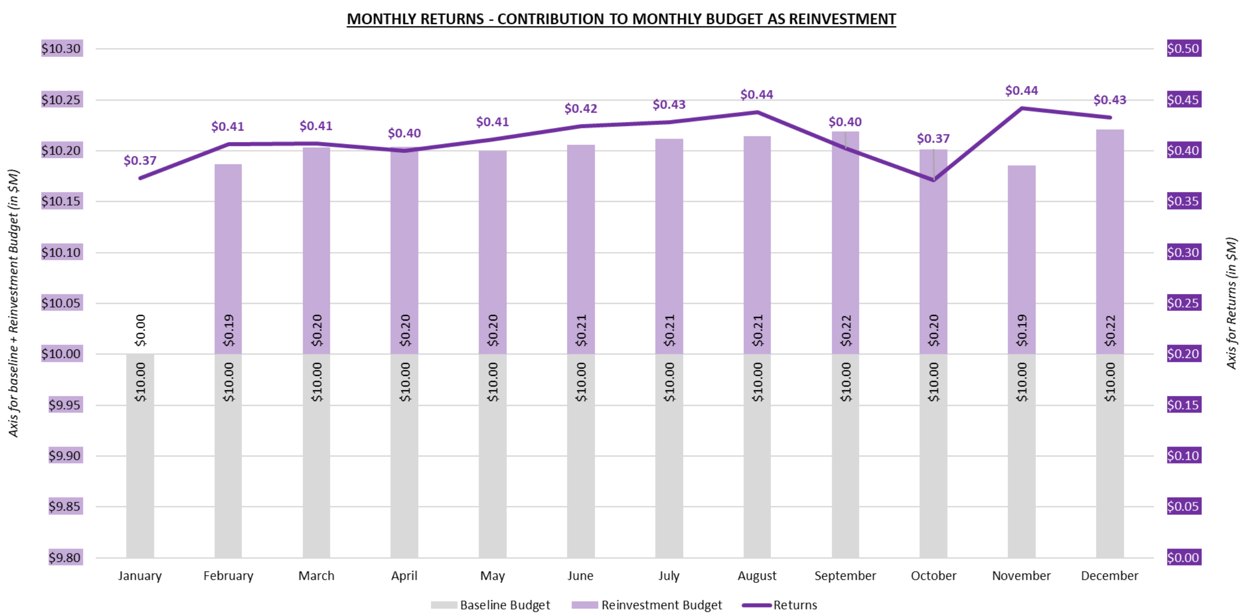
ADVERTISING MEDIUM	CONSTRAINT LOWER BOUND	CURRENT CONSTRAINT	CONSTRAINT UPPER BOUND	OPTIMAL ALLOCATION
Print	-∞%	3.1%	4.9%	$0
TV	3.9%	4.9%	6.2%	$3,000,000
SEO	-∞%	2.4%	3.9%	$0
AdWords	3.3%	3.9%	4.6%	$1,000,000
Facebook	-∞%	1.6%	2.9%	$0
LinkedIn	-∞%	2.4%	3.9%	$0
Instagram	3.9%	4.6%	∞%	$3,000,000
Snapchat	-∞%	2.6%	3.9%	$0
Twitter	-∞%	3.3%	3.9%	$0
Email	2.9%	4.4%	∞%	$3,000,000

Reinvestment of Returns

We now obtained permission to reinvest half of the returns from the previous month as the budget for the next month. This would mean that apart from the baseline budget of $10M, for each month, we are also allowed to reinvest half of the returns from each channel. For this analysis, we again take ROI 1 into consideration.

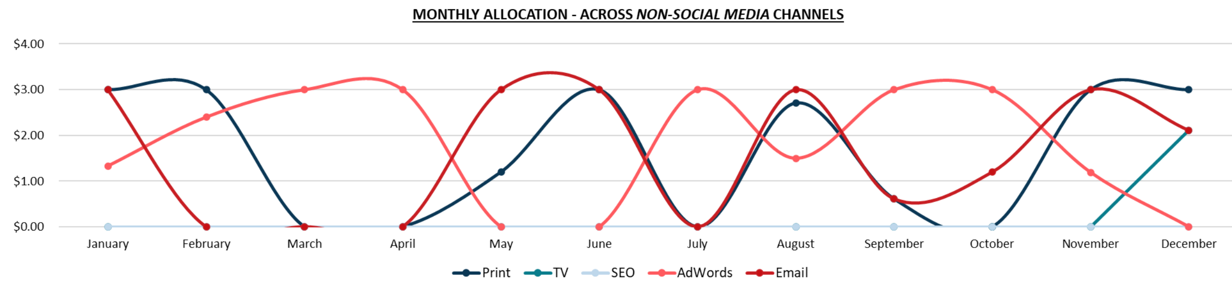
Another consideration is that monthly ROIs for each platform vary and are again estimated and fed into the model.

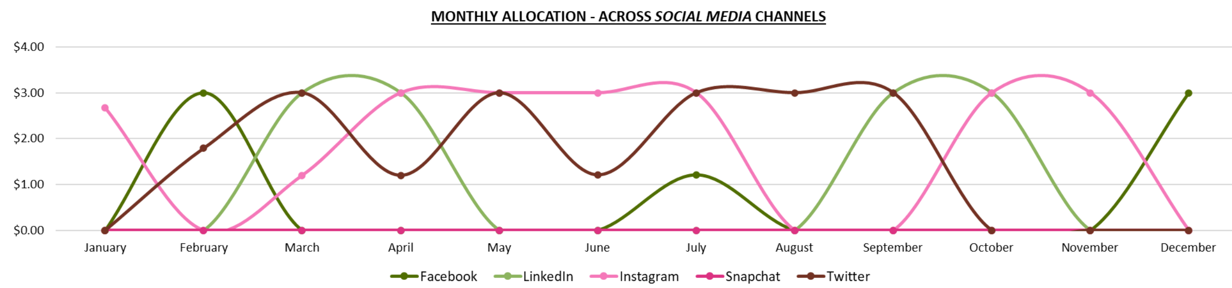
The chart below illustrates the total returns from each month (as a line, in darker shade of purple), which is referred to as the **returns,** and half of this return (as bars, in lighter shade of purple) are reinvested in the next month.



We do not see an upward increasing trend of returns because there is variation in the estimated ROIs at a monthly level. This is contextualized further in the chart(s) below, which denote the movement of allocation per platform per month. There is quite some variability and diversification in the platforms used over the 12-month period.

The charts are split grouping non-social media platforms and social media platforms separately for convenience of viewing.





Identifying a Stable Budget

We define a stable budget as one where the change in monthly allocation for each platform is no more than $1M. The table below lists in each row (for each month), the allocation of the entire budget across the 10 platforms.

It also highlights in red the cells where the monthly change in allocation exceeds $1M.

From the interspersed red cells, we conclude that this budget is **not stable**.

Month	Print	TV	SEO	AdWords	Facebook	LinkedIn	Instagram	Snapchat	Twitter	Email	Monthly Allocation
January	3.000000	0.000000	0.000000	1.330000	0.000000	0.000000	2.670000	0.000000	0.000000	3.000000	10.000000
February	3.000000	0.000000	0.000000	2.400000	3.000000	0.000000	0.000000	0.000000	1.790000	0.000000	10.190000
March	0.000000	0.000000	0.000000	3.000000	0.000000	3.000000	1.200000	0.000000	3.000000	0.000000	10.200000
April	0.000000	0.000000	0.000000	3.000000	0.000000	3.000000	3.000000	0.000000	1.200000	0.000000	10.200000
May	1.200000	0.000000	0.000000	0.000000	0.000000	0.000000	3.000000	0.000000	3.000000	3.000000	10.200000
June	3.000000	0.000000	0.000000	0.000000	0.000000	0.000000	3.000000	0.000000	1.210000	3.000000	10.210000
July	0.000000	0.000000	0.000000	3.000000	1.210000	0.000000	3.000000	0.000000	3.000000	0.000000	10.210000
August	2.710000	0.000000	0.000000	1.500000	0.000000	0.000000	0.000000	0.000000	3.000000	3.000000	10.210000
September	0.610000	0.000000	0.000000	3.000000	0.000000	3.000000	0.000000	0.000000	3.000000	0.610000	10.220000
October	0.000000	0.000000	0.000000	3.000000	0.000000	3.000000	3.000000	0.000000	0.000000	1.200000	10.200000
November	3.000000	0.000000	0.000000	1.190000	0.000000	0.000000	3.000000	0.000000	0.000000	3.000000	10.190000
December	3.000000	2.110000	0.000000	0.000000	3.000000	0.000000	0.000000	0.000000	0.000000	2.110000	10.220000

However, to make it a stable budget, we can explore the following option: adding a new constraint of,

mod(allocation(older model) - allocation(newer model)) <= 10,000,000
[here mod denotes the modulus]

All the other constraints and modeling would remain the same.

CONCLUSION

In conclusion, the results of the optimization problem have shown the optimal allocation of the budget for both cases (ROI 1 & ROI 2). The sensitivity of ROIs were also reported, which can help choose between the two estimates in hand. Given that the optimal allocations were not stable, a solution was also shared to achieve budget stability.

RELEVANT CODE CHUNKS

The following code chunks show

(a) Calculating optimal allocation for First ROI Data with constraints defined:

```
# Initialize a Gurobi model object

q1 = gp.Model()

# Add decision variables (q1X) to the model based on the number of columns in roiData

q1X = q1.addMVar(roiData.shape[1])

# Set the objective function to maximize the sum of ROI multiplied by the respective
decision variable

# The objective function aims to maximize ROI (Return On Investment)

q1.setObjective(gp.quicksum([list(roiData.loc['ROI'])[i] * q1X[i] for i in
range(roiData.shape[1])]), gp.GRB.MAXIMIZE)

# Initialize a list (cons) to store constraint objects

cons = [0] * 4

# Retrieve the names of the media channels from the column headers of roiData

media = list(roiData.columns)

# Add the first constraint: The sum of all decision variables should be less than or
equal to the budget

cons[0] = q1.addConstr(gp.quicksum(q1X[i] for i in range(roiData.shape[1])) <= budget)
```

```
# Add the second constraint: The sum of decision variables for 'Print' and 'TV' must be less than or equal to

# the sum of decision variables for 'Facebook' and 'Email'

cons[1] = q1.addConstr(gp.quicksum(q1X[media.index(x)] for x in ['Print', 'TV']) <=

gp.quicksum(q1X[media.index(x)] for x in ['Facebook', 'Email']))

# List of social media platforms for constraint

socialMedia = ['Facebook', 'LinkedIn', 'Instagram', 'Snapchat', 'Twitter']

# Add the third constraint: The sum of decision variables for socialMedia should be greater than or equal to

# twice the sum of decision variables for 'SEO' and 'AdWords'

cons[2] = q1.addConstr(gp.quicksum(q1X[media.index(x)] for x in socialMedia) >=

2 * gp.quicksum(q1X[media.index(x)] for x in ['SEO', 'AdWords']))

# Add the fourth constraint: Each decision variable should be less than or equal to individualBudget

cons[3] = q1.addConstrs(q1X[i] <= individualBudget for i in range(roiData.shape[1]))

# Run the optimization algorithm to solve the model

q1.optimize()
```

(b) Calculating optimal allocation on a monthly basis:

```
# Set an initial budget amount

initialBudget = 10000000

# Create a list of months based on the index of roiMatData DataFrame

months = list(roiMatData.index)

# Initialize empty lists to store monthly allocation details and ROI

monthsAllocation = []

monthsROI = []

# Initialize variable for storing ROI of each month

monthlyROI = 0

# Initialize list for storing monthly allocation

monthlyAllocation = []

# Loop through each month

for month in months:
```

```
# Print the current month for tracking progress

print("Month: " + month)

# Append a dictionary with month, initial budget, and monthly ROI to monthsAllocation
list

monthsAllocation.append({'Month': month, 'Budget': initialBudget, 'ROI' : monthlyROI})

# Create a Gurobi model for optimization

q5 = gp.Model()

# Add decision variables for the Gurobi model based on the number of columns in
roiMatData for that month

q5X = q5.addMVar(roiMatData.loc[month].shape[0])

# Set the objective function to maximize the sum of ROI multiplied by decision
variables

q5.setObjective(gp.quicksum([list(roiMatData.loc[month])[i]/100 * q5X[i] for i in
range(roiMatData.loc[month].shape[0])]), gp.GRB.MAXIMIZE)

# Initialize a list to hold constraint objects

consQ5 = [0] * 4

# Extract column names to represent different media channels

media = list(roiMatData.columns)

# Add constraints for the Gurobi model

consQ5[0] = q5.addConstr(gp.quicksum(q5X[i] for i in
range(roiMatData.loc[month].shape[0])) <= initialBudget)

consQ5[1] = q5.addConstr(gp.quicksum(q5X[media.index(x)] for x in ['Print', 'TV']) <=

gp.quicksum(q5X[media.index(x)] for x in ['Facebook', 'Email']))

# Define the list of social media platforms

socialMedia = ['Facebook', 'LinkedIn', 'Instagram', 'Snapchat', 'Twitter']

# Add another constraint for social media allocation

consQ5[2] = q5.addConstr(gp.quicksum(q5X[media.index(x)] for x in socialMedia) >=

2* gp.quicksum(q5X[media.index(x)] for x in ['SEO', 'AdWords']))

# Add constraints for individual budgets for each channel

consQ5[3] = q5.addConstrs(q5X[i] <= individualBudget for i in
range(roiMatData.loc[month].shape[0]))

# Run the optimization
```

```
q5.optimize()

# Store the optimized objective value as the monthly ROI

monthlyROI = q5.ObjVal

# Append monthly allocation details

monthlyAllocation.append({'Month': month, 'Allocation': list(q5.X) +
[sum(list(q5.X))]})

# Reset the initial budget to $10,000,000

initialBudget = 10000000

# Update the budget based on the obtained monthly ROI, only if ROI is positive

if monthlyROI >= 0:

initialBudget = initialBudget + 0.5*monthlyROI

else:

# Print a message if the monthly ROI is negative

print("Monthly ROI for " + month + " is negative.")
```