The Perfect Boat Cruise

Problem Description

I would be working on a real-world scenario where convex optimization, optimization techniques, and mixed integer programming could be applied. Although this is a variant of the knapsack problem, this problem is one which I have come to know as the Perfect Boat Cruise Problem.

Imagine you just bought a new ship, and are about to open this ship to the public for its first ride. As a new venture, you wish to maximize revenue from this ride, avoid as much damage to the ship as possible, and increase the popularity of your boat in the best way possible. Here, I explore ways to optimize the selection of your first set of customers, such that you, as the cruise owner, gets the most out of this ride.

This problem isn't one which is only applicable to ship cruises. Given the spike in celebrity status, and increase in demand for tourism, the solution to this optimization problem could apply to the opening and use of other exotic experiences such as plane trips, excursions and events to the public, such that the organizers make as much money as possible, avoid as much damage as possible, and still allow for a spread of the word through this event. Building owners, for example, who wish to increase publicity for their rental business could use results of this analysis to determine the best individuals to rent apartments in their buildings to during periods of high demand, such that they maximize revenue and reduce damages. With respect to the tech world, results from this analysis could even provide insight into how to optimize ad revenue and outreach for your site, while maintaining an optimized site for a google search.

Solution Specification

I begin this analysis with a very basic model and go on later in the paper to test out much more interesting variants of the same problem. In the simpler case, the boat owner simply wishes to maximize only the revenue he would be generating from his cruise. Constraints for this base case problem could include constraints on

- the number of kids allowed during the trip (to reduce damage on the ship),
- the least number of celebrities expected to participate in the cruise (celebrities increase publicity)
- The maximum total weight allowed on the ship

Based on the above information, one could express the maximization problem as

$$\text{maximize} \sum_{1}^{N} B^T{}_N \cdot X_N$$

$$\text{subject to} \sum_{1}^{N} W^T{}_N \cdot X_N <= maxweight$$

$$\sum_{1}^{N} K^T{}_N \cdot X_N <= maxkids$$

$$\sum_{1}^{N} C^T{}_N \cdot X_N >= mincel$$

$$X = \{0, 1\}$$

Where B represents the bidding by each individual for the boat ticket, K represents the number of kids each individual has, C is a boolean representing the individual’s celebrity status, and X is the boolean variable to be optimized; where 1 means that an individual goes on the cruise.

The above maximization problem is a nonconvex programming problem, and a mixed integer optimization problem(with binary variables). This problem is disqualified as an LP, since the feasible set for a linear programming problem should always be a convex function, and the X = {0,1} constraint of this given problem is neither a continuous nor convex function.

From the above pieces of information, one could tell that solving this optimization problem manually could involve a technique such as the branch and bound method. Branch and bound method involves a tree-like search, where branches representing the subsets of the solution set are visited, and compared with bounds for the optimal solution. A particular branch is then taken only if it can produce a better solution than the previously calculated optimal solution.

Analysis

For the analysis, 700 pieces of random data were generated and used for the optimization tests based on a series of cvxpy algorithms. This data included the following measures

- Name of the individual
- Age of the individual
- Number of kids
- Celebrity status(boolean)
- The weight of the individual
- Amount bid on the cruise
- Gender

This data was then organized, and fed into a cvxpy algorithm which optimizes the amount of money made by the boat owner, while satisfying certain constraints on the number of kids allowed on the cruise(<60), number of celebrities needed(>10), and the maximum weight the boat can carry(1000 kilos).

Running the algorithm on the random data, we get the result below

```
['Name', 'Age', 'Weight', 'Gender', 'Bid', 'Kids', 'Celebrity']
[['Alberto Lapierre' '32' '46' '0' '9033' '1' '1']
 ['Martha Cobos' '29' '53' '1' '12912' '4' '0']
 ['Barbara Tewksbury' '25' '46' '1' '9359' '1' '1']
 ['Jorge Porter' '41' '44' '1' '13619' '4' '0']
 ['William French' '29' '62' '1' '13533' '3' '1']
 ['Jeremy Vise' '76' '42' '0' '9166' '3' '1']
 ['Vanessa Dupont' '32' '41' '0' '8755' '3' '1']
 ['Lisa Serrano' '77' '43' '1' '7901' '1' '0']
 ['Cheryl Debruyn' '29' '41' '0' '13557' '3' '0']
 ['Kenneth Medel' '33' '55' '1' '13716' '4' '1']
 ['James Coutts' '76' '45' '1' '8378' '0' '1']
 ['James Oliveira' '40' '40' '1' '13783' '4' '1']
 ['Bonnie Smith' '45' '49' '0' '13305' '4' '0']
 ['Ella Smith' '34' '53' '0' '12673' '4' '0']
 ['Milo Nau' '55' '47' '0' '13691' '3' '1']
 ['Kathryn Emmons' '32' '64' '1' '13292' '4' '0']
 ['Leann Henning' '51' '40' '0' '8268' '0' '0']
 ['Jose Rader' '36' '47' '0' '9873' '1' '1']
 ['Jason Watkins' '72' '40' '1' '7192' '0' '1']
 ['Frances Vargas' '39' '58' '1' '12984' '3' '0']
 ['Phyllis Juneau' '31' '44' '0' '9500' '0' '0']]
21 people were selected with a total weight of  1000 and a total of 50  kids and 11 celebrities
Your total revenue is  234490
```

This algorithm, in fact, works and solves the problem as we would expect (when you inspect the results).

We would like to push this analysis further and attempt to model as much of the real world scenario as possible. It is obvious that profit isn't usually the only ingredient of interest. The shipowner could have different levels of interests in achieving different objectives, including making profit, reducing damage, and getting celebrity recommendation. In one case, a company might just be focused on gaining celebrity recognition, while another company could be equal parts reducing damage and gaining profit. Given that this is the case, there could exist 3 or more objective functions, which have been weighted according to their level of importance to the shipowner. This would, in fact, make the problem a multi-objective-multiple integer optimization problem.

To go about optimizing this variant of the cruise problem,, it is essential to normalize the data(through centering and scaling) in order to reduce the effect of differences in scales between the different objective units. After the scaling is done, a maximization problem with 3 weighted objective functions is solved, subject to the weight constraint. The 3 components of this multi-objective function are

1. Maximize the total revenue generated * an importance level of 0.6
2. Maximize the number of celebrities attending * a weight of 0.2
3. Minimize the amount of damage/kids * weight of 0.2 (or maximize -damage*0.2)

Running the code for this new multi-objective optimization problem, we find a completely different result as in the earlier case, as shown below;

```
['Name', 'Age', 'Weight', 'Gender', 'Bid', 'Kids', 'Celebrity']
[['Mary Johnson' '36' '73' '1' '12446' '3' '1']
 ['Jorge Porter' '41' '44' '1' '13619' '4' '0']
 ['Richard Terry' '56' '90' '0' '13368' '3' '1']
 ['William French' '29' '62' '1' '13533' '3' '1']
 ['Warren Hibbard' '78' '93' '1' '13676' '3' '1']
 ['Hilda Wesley' '28' '77' '1' '13473' '3' '1']
 ['Cheryl Debruyn' '29' '41' '0' '13557' '3' '0']
 ['Kenneth Medel' '33' '55' '1' '13716' '4' '1']
 ['James Oliveira' '40' '40' '1' '13783' '4' '1']
 ['Bonnie Smith' '45' '49' '0' '13305' '4' '0']
 ['Milo Nau' '55' '47' '0' '13691' '3' '1']
 ['Herbert Garofalo' '18' '64' '1' '13194' '4' '1']
 ['Sandra Lyons' '46' '70' '0' '13223' '4' '1']
 ['Jose Rader' '36' '47' '0' '9873' '1' '1']
 ['Charles Beauchemin' '41' '77' '1' '13721' '3' '1']
 ['Lola Wright' '50' '69' '0' '13887' '3' '0']]
16 people were selected with a total weight of  998 and a total of 52  kids and 12 celebrities
Your total revenue is  212065
```

An even more solidified real-world replica of this scenario is one in which a shipowner has a strict upper or lower bound on one or all of these various, and still wishes to achieve each of these different objectives according to personal weights. To push through with this, it would be necessary to also normalize the constraints, in order to make them suitable for use with the normalized data. Observing the results, we even have another unique, and interesting solution to the problem based on the algorithm.

```
['Name', 'Age', 'Weight', 'Gender', 'Bid', 'Kids', 'Celebrity']
[['Alberto Lapierre' '32' '46' '0' '9033' '1' '1']
 ['Barbara Tewksbury' '25' '46' '1' '9359' '1' '1']
 ['Richard Terry' '56' '90' '0' '13368' '3' '1']
 ['William French' '29' '62' '1' '13533' '3' '1']
 ['Warren Hibbard' '78' '93' '1' '13676' '3' '1']
 ['Terrence Irizarry' '42' '85' '0' '9659' '0' '1']
 ['Hilda Wesley' '28' '77' '1' '13473' '3' '1']
 ['Cheryl Debruyn' '29' '41' '0' '13557' '3' '0']
 ['Kenneth Medel' '33' '55' '1' '13716' '4' '1']
 ['James Oliveira' '40' '40' '1' '13783' '4' '1']
 ['Milo Nau' '55' '47' '0' '13691' '3' '1']
 ['Deangelo Duell' '76' '56' '0' '9833' '1' '1']
 ['Jose Rader' '36' '47' '0' '9873' '1' '1']
 ['Charles Beauchemin' '41' '77' '1' '13721' '3' '1']
 ['Phyllis Juneau' '31' '44' '0' '9500' '0' '0']
 ['Viola Sigrist' '76' '49' '0' '9437' '1' '1']]
16 people were selected with a total weight of  955 and a total of 34  kids and 14 celebrities
Your total revenue is  189212
```

Meta-analysis of the 3 Methods so far

	Basic Solution	Weighted Solution for multi-objective optimization	Combined approach
Structure	$maximize \sum_1^N B^T{}_N \cdot X_N$ $subject\ to \sum_1^N W^T{}_N \cdot X_N <= maxweight$ $\sum_1^N K^T{}_N \cdot X_N <= maxkids$ $\sum_1^N C^T{}_N \cdot X_N >= mincel$ $X = \{0, 1\}$	$maximize \sum_1^N B^T{}_N \cdot X_N \cdot w_1 + K^T{}_N \cdot X_N \cdot w_2$ $+ C^T{}_N \cdot X_N \cdot w_3$ $subject\ to \sum_1^N W^T{}_N \cdot X_N <= maxweight$ $X = \{0, 1\}$	$maximize \sum_1^N B^T{}_N \cdot X_N \cdot w_1 + K^T{}_N \cdot X_N \cdot$ $+ C^T{}_N \cdot X_N \cdot w_3$ $subject\ to \sum_1^N W^T{}_N \cdot X_N <= maxweight$ $\sum_1^N K^T{}_N \cdot X_N <= maxkids$ $\sum_1^N C^T{}_N \cdot X_N >= mincel$ $X = \{0, 1\}$
Revenue	234490	212065	189212
Kids onboard	50	52	34
Adults onboard	21	16	16
Celebrities onboard	11	12	14
Total weight	1000	998	955

Summary/Self-Reflection

Using the cvxpy framework, we have seen how the results of various approaches to the same problem look like. Based on these results, we could expect that a boat owner who is more lax, and wishes to just constrain all other factors and maximize profit would make the highest amount of profit, but might be missing out on the possibility of higher publicity due to the nature of his approach. A cruise owner who just uses weights to determine client selection would make a large amount of profit too and seems to have the most evenly satisfied objectives. The final cruiser, who implements a combination of both methods would make less profit compared to his peers but would have higher publicity, and space for miscellaneous weights; given the inputs.

In summary, optimizing profit for a space leasing company could take a multi-objective multi-integer approach, a basic approach, or a combination of both; and using the above techniques, owners of such spaces can model, and even optimize the selection of clients for such spaces - to optimize for whatever they deem most important.

References

Branch and bound. (2018). Retrieved from https://en.wikipedia.org/wiki/Branch_and_bound

Knapsack problem. (2018, December 18). Retrieved from https://en.wikipedia.org/wiki/Knapsack_problem

Helmut M. (2006). Normalization and Other Topics in Multi-Objective Optimization. Retrieved on December 21, 2018

Problems¶. (n.d.). Retrieved from https://www.cvxpy.org/api_reference/cvxpy.problems.html

Appendix

Generating the random data

```
import numpy as np
import names
import random
import cvxpy as cp
import numpy.linalg as nl

naames ,aages, weight, gender, bid ,kids , celeb = [],[], [], [], [],[],[]

for i in range(600):
    naames.append(names.get_full_name())
    aages.append(random.randint(18, 80))
    weight.append(random.randint(40, 200))
```

```
    gender.append(random.randint(0, 1))
    t = random.randint(0, 4)
    kids.append(t)
    celeb.append(random.randint(0, 1))
    if t > 2:
        bid.append(random.randint(8000, 14000))
    else:
        bid.append(random.randint(4000, 10000))

naames = np.array(naames)
aages = np.array(aages)
weight = np.array(weight)
gender = np.array(gender)
bid = np.array(bid)
kids = np.array(kids)
celeb = np.array(celeb)

holder= []

for i in range(600):
    holder.append([naames[i], aages[i], weight[i], gender[i], bid[i],
kids[i], celeb[i]])
```

Applying the first method

```
import math

X = cp.Variable(len(weight), boolean = True, integer = True)
cash = bid.T*X

constraints = [weight.T*X <= 1000, kids.T*X <= 50, celeb.T*X >= 10]
prob = cp.Problem(cp.Maximize(cash), constraints)
result = prob.solve()
```

```
indices = []

for i in range(600):
    if round((X.value[i]),5) >= 1.0:
        indices.append(i)
```

```
solution = np.array([holder[i] for i in indices])

print(["Name", "Age", "Weight", "Gender", "Bid", "Kids", "Celebrity"])
print(solution)

totalkids, totalweight ,totalc ,totalcash = 0,0,0,0
nweight = []
nbid = []
for i in solution:
    totalkids += int(i[5])
    totalweight += int(i[2])
    totalcash += int(i[4])
    totalc += int(i[-1])
    nweight.append(i[2])
    nbid.append(i[4])
print(len(solution), "people were selected with a total weight of ",
totalweight, "and a total of", totalkids, " kids and", totalc
,"celebrities")
print("Your total revenue is ", totalcash)
```

Applying the second method

```
import sklearn.preprocessing as sk
import math

X = cp.Variable(len(weight), boolean = True, integer = True)

cash = sk.scale(bid).T*X
popularity = sk.scale(celeb).T*X
damage = sk.scale(kids).T*X

constraints = [weight.T*X <= 1000]
prob = cp.Problem(cp.Maximize(0.6*cash + 0.2*popularity -0.2*damage),
constraints)
result = prob.solve()
```

```
indices = []

for i in range(600):
    if round((X.value[i]),5) >= 1.0:
        indices.append(i)

solution = np.array([holder[i] for i in indices])

print(["Name", "Age", "Weight", "Gender", "Bid", "Kids", "Celebrity"])
print(solution)

totalkids, totalweight ,totalc ,totalcash = 0,0,0,0
nweight = []
nbid = []
for i in solution:
    totalkids += int(i[5])
    totalweight += int(i[2])
    totalcash += int(i[4])
    totalc += int(i[-1])
    nweight.append(i[2])
    nbid.append(i[4])
print(len(solution), "people were selected with a total weight of ",
totalweight, "and a total of", totalkids, " kids and", totalc
,"celebrities")
print("Your total revenue is ", totalcash)
```

Applying the third method

```
import sklearn.preprocessing as sk
import math

X = cp.Variable(len(weight), boolean = True, integer = True)

cash = sk.scale(bid).T*X
popularity = sk.scale(celeb).T*X
damage = sk.scale(kids).T*X

constraints = [weight.T*X <= 1000, sk.scale(kids).T*X <=
(50-np.mean(kids))/nl.norm(kids, 2), sk.scale(celeb).T*X >=
(10-np.mean(celeb))/nl.norm(celeb, 2)]
prob = cp.Problem(cp.Maximize(0.6*cash + 0.2*popularity -0.2*damage),
```

```
constraints)
result = prob.solve()
```

```
indices = []

for i in range(600):
    if round((X.value[i]),5) >= 1.0:
        indices.append(i)

solution = np.array([holder[i] for i in indices])

print(["Name", "Age", "Weight", "Gender", "Bid", "Kids", "Celebrity"])
print(solution)

totalkids, totalweight ,totalc ,totalcash = 0,0,0,0
nweight = []
nbid = []
for i in solution:
    totalkids += int(i[5])
    totalweight += int(i[2])
    totalcash += int(i[4])
    totalc += int(i[-1])
    nweight.append(i[2])
    nbid.append(i[4])
print(len(solution), "people were selected with a total weight of ",
totalweight, "and a total of", totalkids, " kids and", totalc
,"celebrities")
print("Your total revenue is ", totalcash)
```