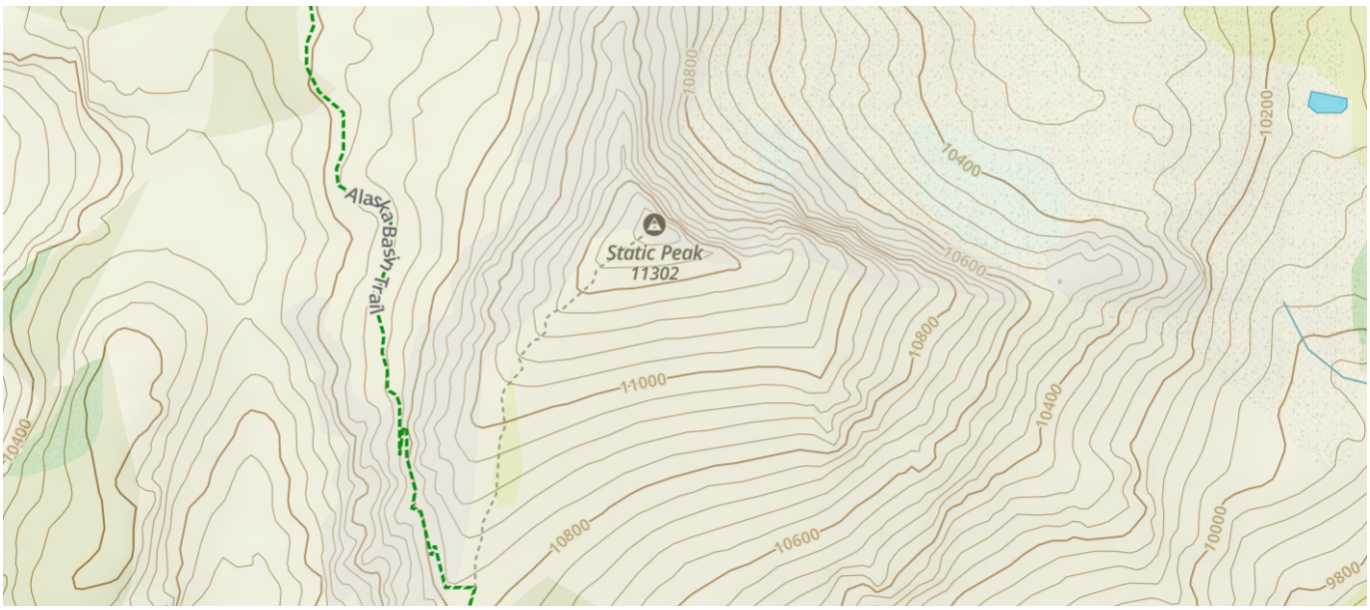
Hw1 Write up

Jacob Child

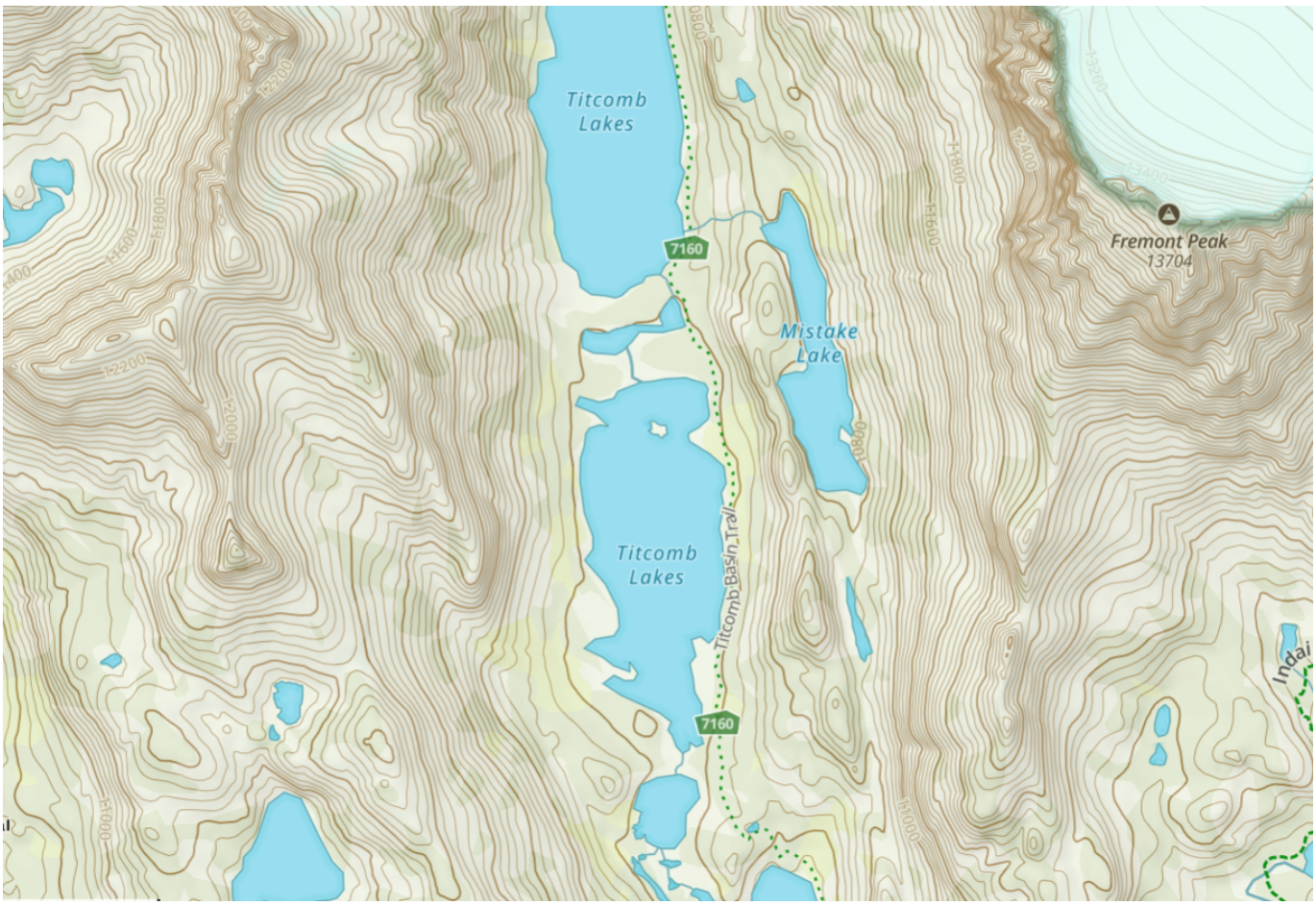
Jan 21st, 2025

1.1: Familiarization with contours and constraints

Problem 1: Topographical examples of possible optimization topologies

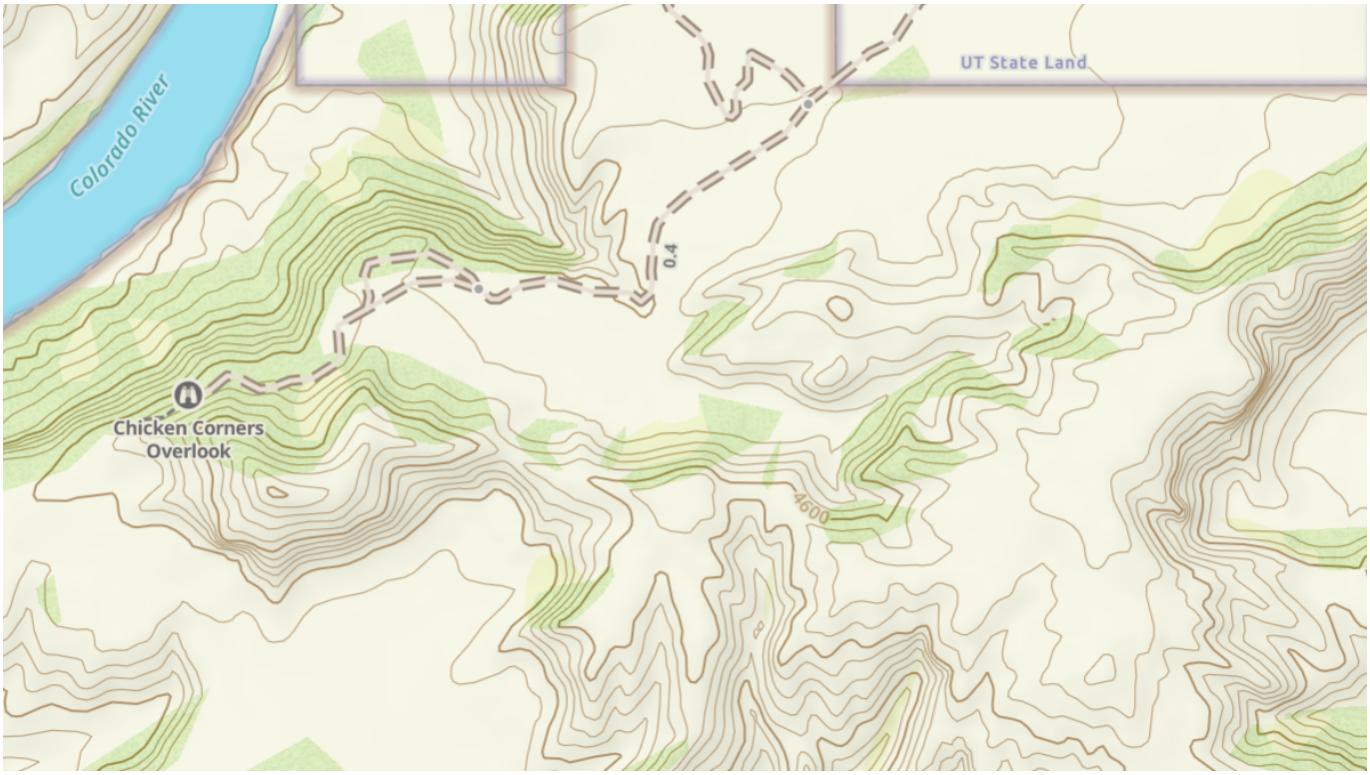


a - Mountain Peak: Static peak (43.68274, -110.81627), Grand Tetons, have been

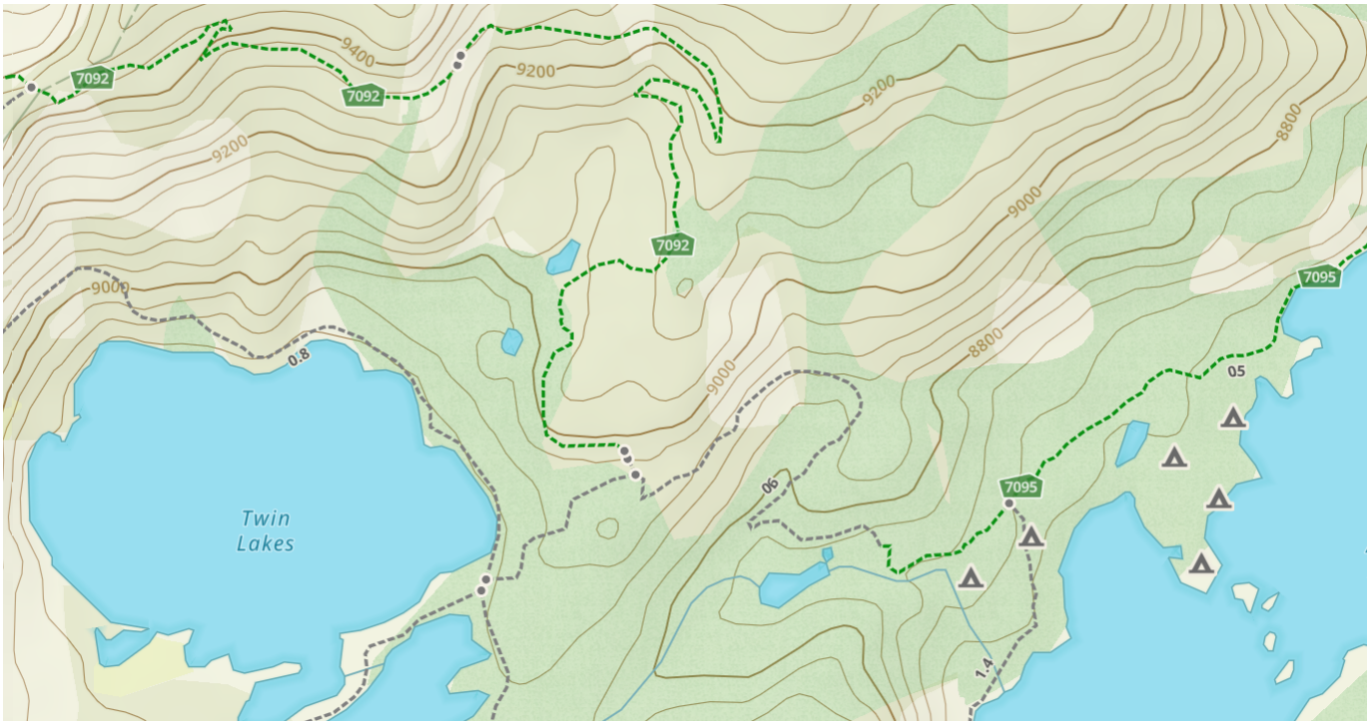


b- Mountain basin or bowl: Titcomb lakes (43.10894, -109.63543), Wind River Range Wyoming, have been

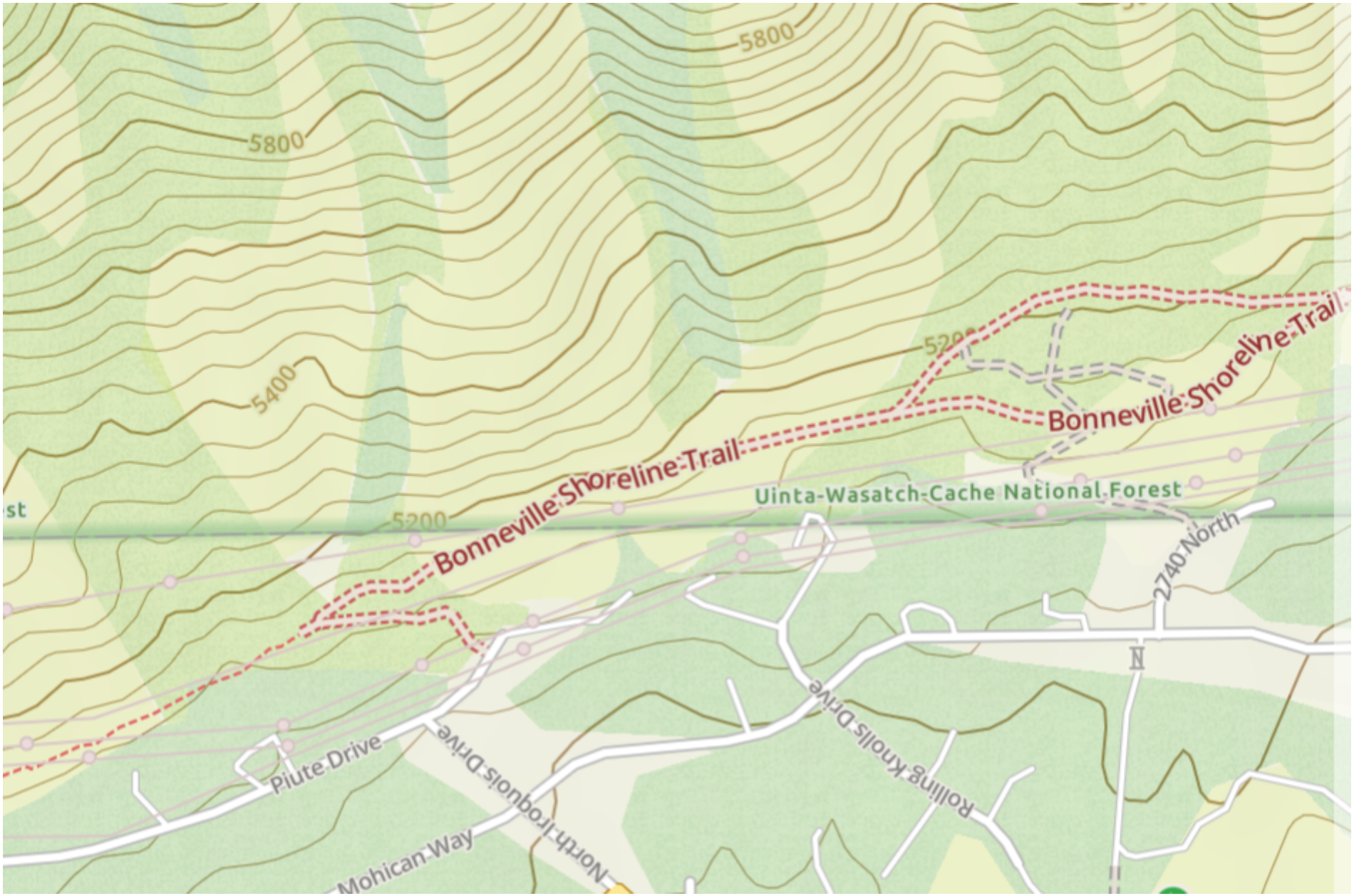
c - A long valley: Chilkoot trail/ Happy camp (59.73788, -135.17756) and long lake, Canada, have been there



d - A saddle: Chicken Corners Overlook, Moab, just to the right in the image is a saddle (38.44098, -109.73116). I've been to the overlook I believe, but not the saddle.



e - Equality constraint: The trail (43.94052, -114.95353), Alice and Toxaway Loop, is an equality constraint, you have to be on the trail to be on the trail :) Sawtooth Mountains Idaho (have been there)


5800
5800
5400
5200
5200
Bonneville Shoreline Trail
Bonneville Shoreline Trail
Bonneville Shoreline Trail
Uinta-Wasatch-Cache National Forest
2740 North
N
Piute Drive
North Iroquois Drive
Rolling Knolls Drive
Mohican Way

f - inequality constraint: Around Bonneville Shoreline trail, Provo. (40.27889, -111.63433) I can't remember where exactly, but some friends and I were exploring, and when we tried to come down we came to a fence for an orchard (I think) and had to follow it for a while.

1.2 Unconstrained Brachistocrone problem

A.) Plot the optimal shape for N = 12

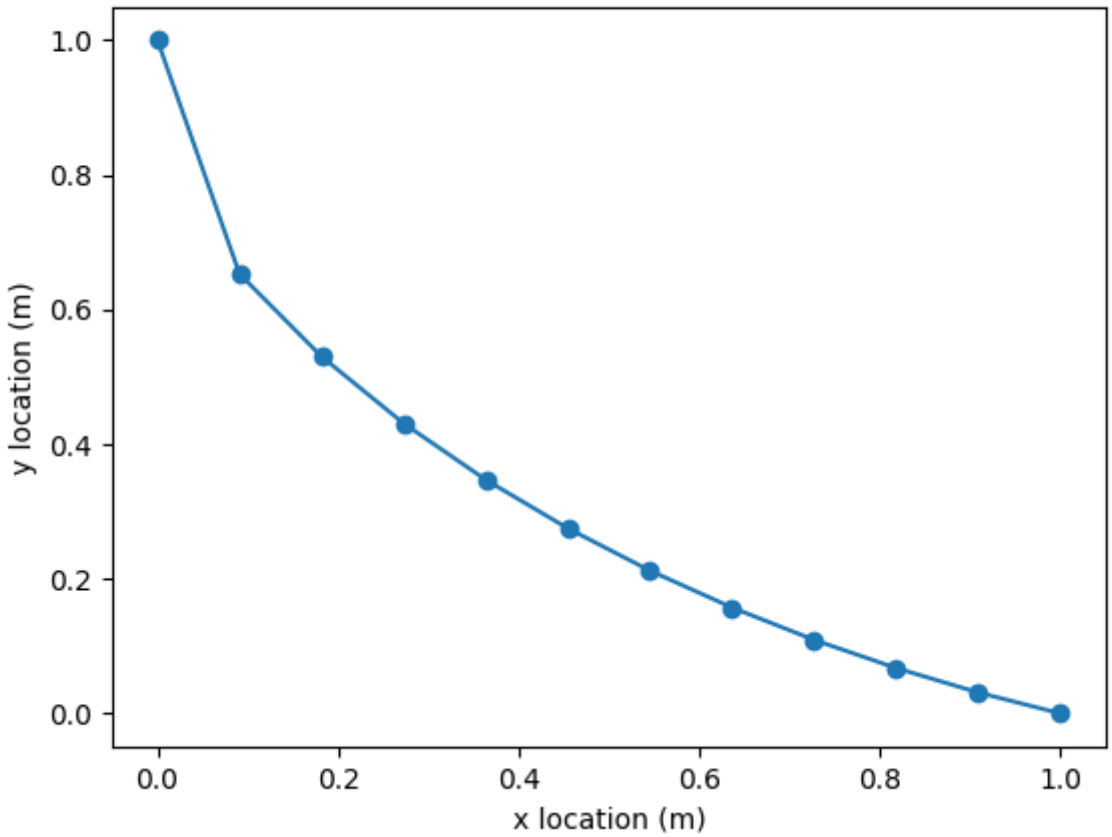


Figure 1.2.1: The path of the N=12 optimization for the minimal travel time of the bead

B.) In a table report travel time when n = 12

N	Travel Time (s)
12	0.603

C.) Increased Dimensionality problem (function calls and wall time)

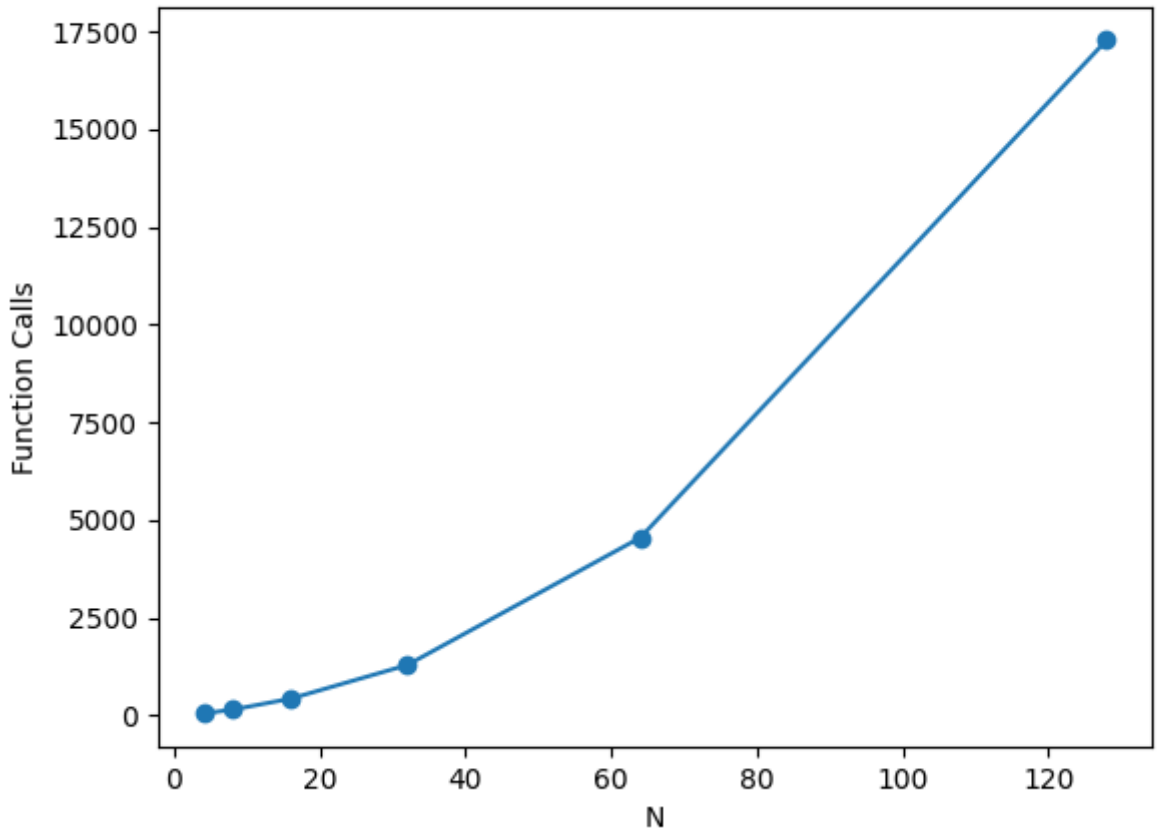


Figure 1.2.2: Function calls vs number of design variables. The number of function calls seems to increase almost exponentially with the number of design variables. It would be important to balance the needs/gain from extra design variables with the exponential computational cost.

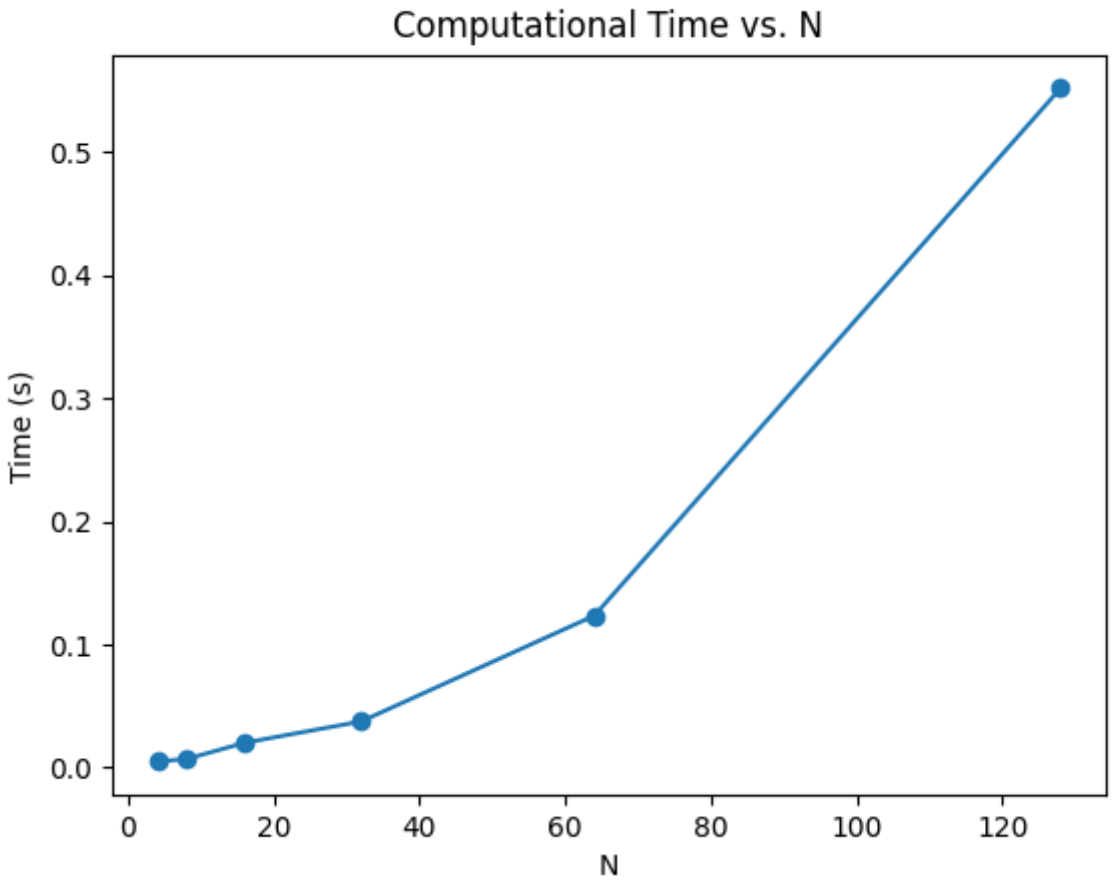


Figure 1.2.3: Time to calculate vs number of design variables. Note that all of these were warm started off of an interpolation of the lower dimension solution. It too appears to be exponential and a linear scaling with function calls.

N	Ball Time (s)
4.0	0.6364934027024898
8.0	0.6102776427778198
16.0	0.598897476939211
32.0	0.592971974936848
64.0	0.5896385652296386
128.0	0.5876904166193399

D.) What I learned

The complexity of the design can exponentially scale the computational costs of optimizing the problem, even with warm starting. Also, the gains at higher dimensionality in this case were quite small- 0.5896 seconds to 0.5877 seconds when doubling the number of design variables from 64 to 128.

1.3 Constrained Truss Problem

A.) Optimal Mass and corresponding cross sectional areas

Mass = 1497.60 lbs

Cross Sectional Areas:

Member	Cross Sectional Area (in^2)
1.0	7.900000000058272
2.0	0.10000000005790924
3.0	8.100000000058003
4.0	3.900000000057788
5.0	0.10000000005805135
6.0	0.10000000005798029
7.0	5.7982756058117415
8.0	5.515432893337165
9.0	3.6769552622523425
10.0	0.14142135631941202

B.) Convergence Plot

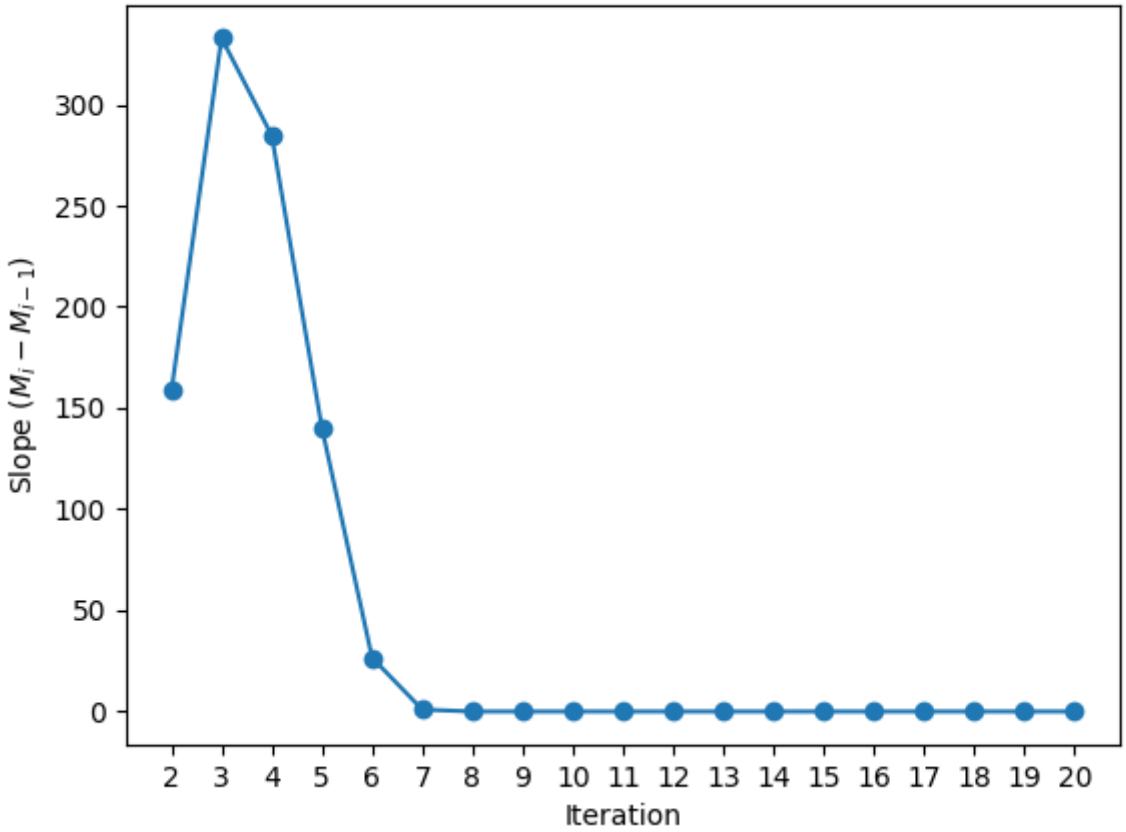


Figure 1.3.1: This is a plot of the slope of convergence, or the difference between the current mass and the last iteration mass. It shows that after iteration 7, the changes in the overall mass were very small, and the slope converges towards 0, showing first order optimality. It should be noted that these "iterations" are actually the number of times the Jacobian was calculated and the callback function called. The optimizer reported 24 iterations, but the Jacobian and callback function were only ran/called 20 times

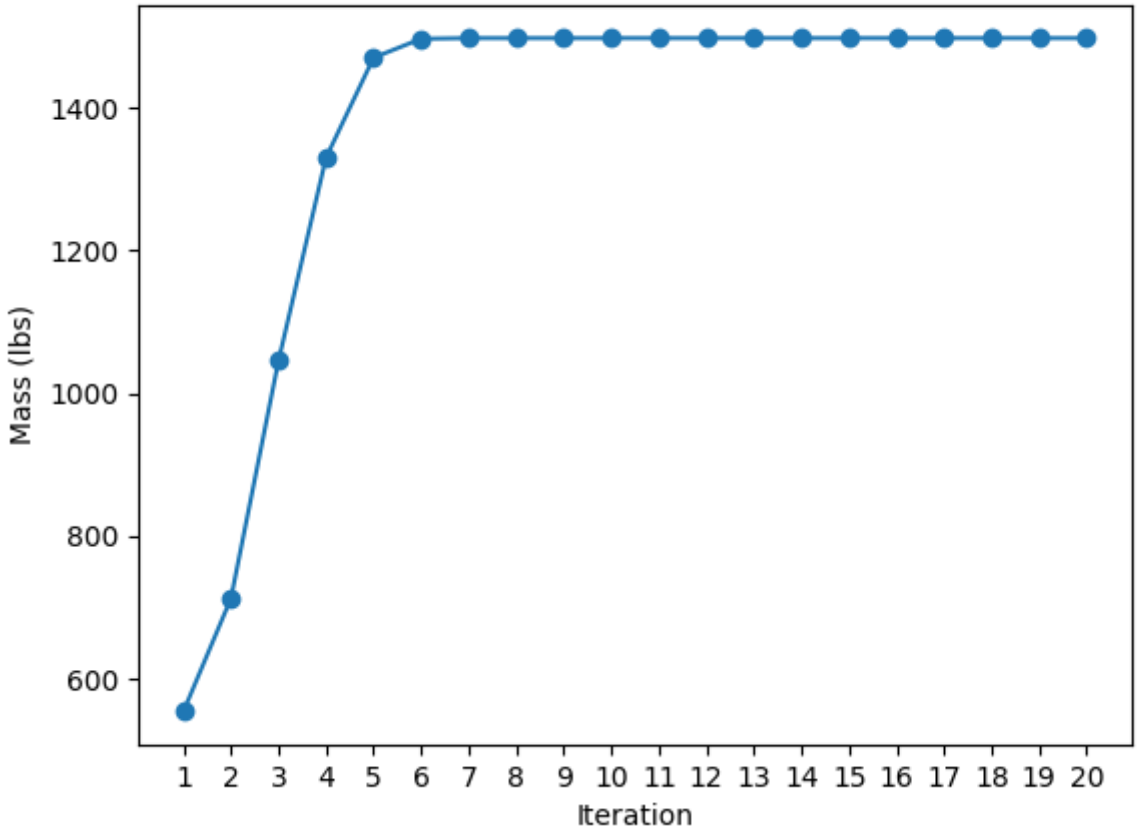


Figure 1.3.2: This is a plot of the final mass found by the optimizer over iterations. You'll note that it appears to actually be maximizing instead of minimizing, this is because it is trying to satisfy the stress constraints. When starting with really large initial values, it will take a big step down, and then rise up again similar to the plot shown. In decreasing mass it quickly decreases cross-sectional area which results in yielding and needs to be increased again.

C.) Number of Function Calls to Converge

Number of Function Calls (of the truss function): 298

D.) What I learned

I learned several things- first, that I didn't fully know what the optimizer was going to do or how, and that its behavior was unexpected. Because the constraint runs after the bounds (I think), it immediately tried to shrink the areas as small as possible and then had to increase them again. I also learned how quickly it converges to a solution that is quite good, then over half of the iterations were for more marginal gains.