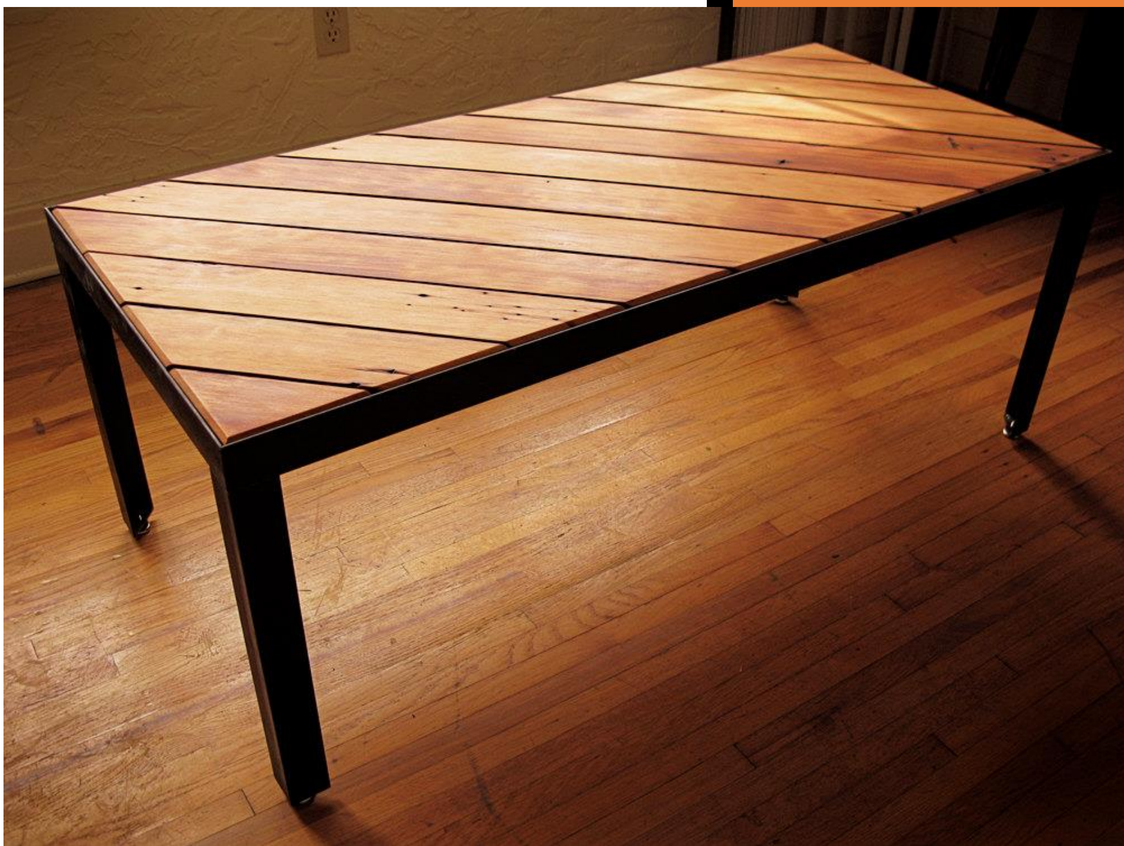
MSE 426

Design Optimization Project Report



Devpreet Bhullar 301217638

Ruicheng Xu 301132013

Refayet Siam 301210102

Table of Contents

Introduction 1
Optimizing the center distance of a spur gear train to reduce backlash. 2
By Refayet Siam 2
Objective function 3
Constraints 3
Best defensive Build in DOTA 4
By Ruicheng xu 4
Assumptions 4
Preset value List: 4
Variables 4
Objective function 5
Constraints 7
Design Optimization of a Table 7
By Devpreet Bhullar 7
Objective function 8
Constraints 8
Method Selection 9
Solution Report 9
Analysis 14
Conclusion 15
References 16

Table of Figures

Figure 1: Spur gears to be optimized ....
Figure 2: Damage reduction due to armor 6
Figure 3: Counter-Clockwise from the top: a) Top view of the table. 2) Front view of the table. 3) Side view of the table. 8
Figure 4: Bar chart comparing optimized variables values produced from fmincon 10
Figure 5: Bar chart comparing optimized variables values produced from GA 11
Figure 6: Dimensions obtained using fmincon: Counter-Clockwise from the top: a) Top view of the table. 2) Front view of the table. 3) Side view of the table. 12
Figure 7: Dimensions obtained using GA: Counter-Clockwise from the top: a) Top view of the table. 2) Front view of the table. 3) Side view of the table 13
Figure 8: LEFT: Minimized volume comparison for different fmincons. RIGHT: Minimized volume comparison for different GA solutions 14
Figure 9: Minimized Volume comparison between fmincon and GA 15

Introduction

Design optimization is the process of finding the optimal variables which yield maximum or minimum value of an objective function subjected to specific requirements called constraints. This report contains solution and analysis for a nonlinear optimization problem selected from a set of three problems where each problem was formulated by a group member. The problems stated in this report are formulated step by step taking various linear and non-linear constraints into account. After formulation, the different problems were analyzed using MATLAB optimization toolbox solvers *'fmincon'* and *'ga'* for the optimization problem.

Optimizing the center distance of a spur gear train to reduce backlash.

By Refayet Siam

One of my personal project was designing a Vertical Take-Off Landing (VTOL) Autonomous Unmanned Aerial Vehicle (AUAV). The idea was to take off vertically like a helicopter and rotate the wings gradually to transition to a fixed wing aircraft mode. This required a gear box to rotate the wing. I chose to design a spur gear train for the problem. Even though worm gears would be ideal in controlling position, they are not very easy to print using 3D printers. I decided to 3D print my parts to completely customize the design and go through various iterations very fast. This saved time and money lost during shipping.

The first iteration of the spur gear train introduced massive backlash. Backlash is a very common problem in spur gear design. It is a clearance between mating gear teeth due to the lack of contact amongst the gear teeth. This can be a good thing as this provides space for lubrication and prevents overheating and tooth damage, however, for precision control, this results in loss of resolution in motion. The backlash can be reduced by either spring loading the gears or designing the gears with a smallest possible center distance. As design was already too compact for spring loading, I decided to minimize the center distance of the spur gears 1 and 2 from the figure below.

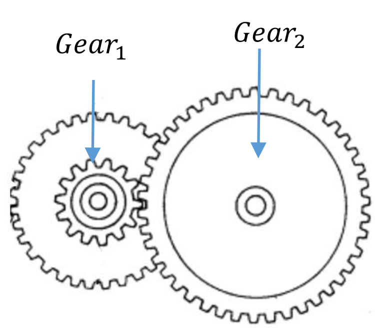


Figure 1: Spur gears to be optimized

Objective function

The center distance comprises of the two diametral pitches, pitch diameter and the number of gear teeth of the two gears. The variables chosen are essential and the only ones that can be varied given that the power output, material, geometric constant and face-width are already fixed. Given these, determining the center distance of two gears and gives the following cost function:

$$f(N_1, N_2, D_1, D_2, P) = \frac{\frac{N_1}{D_1} + \frac{N_2}{D_2}}{2P}$$

Constraints

One of the biggest constraints of the gear design is the yield strength of the material used for 3D printing. I will be using polylactic acid (PLA) to print the CAD modelled gears. The Yield of PLA is 18.5 Mpa and therefore, applying the constraint forms the constraint equations for $gear_1$ and $gear_2$.

$$\boldsymbol{g_1} = Bending\ Stress\ on\ gear_1;\ \ \sigma_{b_1} = \frac{W_{tP_{d1}}}{FJ} < 18.5\ Mpa$$

$$\boldsymbol{g_2} = Bending\ Stress\ on\ gear_2;\ \ \sigma_{b_1} = \frac{W_{tP_{d1}}}{FJ} < 18.5\ Mpa$$

The tangential force on both the gears are the same i.e.:

$$W_{t1} = \frac{H}{\frac{n_1 N_1}{d_{P1}}} = W_{t2} = \frac{H}{\frac{n_2 N_2}{d_{P2}}}$$

We know that,

$$\frac{W_{t2}}{W_{t1}} = \frac{N_2}{N_1}$$

Therefore the third constraint comes out as a non-linear constraint:

$$\boldsymbol{g_3} = \frac{\frac{n_1 N_1}{d_{P1}}}{\frac{n_2 N_2}{d_{P2}}} < \frac{N_2}{N_1}$$

$W_t = tangential\ force$

$P_{d1} = pitch\ diameter\ of\ gear_1\ (mm)$

$P_{d2} = pitch\ diameter\ of\ \ gear_2\ (mm)$

$F = face\ width\ (mm)$

$J = geometric\ constant$

$W_t = tangential\ force$ (N)

$n_1 = revolutions\ per\ minute\ gear_1$

$n_2 = revolutions\ per\ minute\ gear_2$

$N_1 = no\ of\ teeth\ gear_1$

$N_2 = no\ of\ teeth\ gear_2$

$d_{P1} = diametral\ pitch\ gear_1\ (mm)$

$d_{P2} = diametral\ pitch\ gear_2\ (mm)$

Best defensive Build in DOTA

By Ruicheng xu

In this project, I applied knowledge I have learnt from 426 class to optimize a practical problem. How to build a best defensive build for a hero in the DOTA2. The idea of the optimization project is build a tool for the beginner player to get a correct build for the game in a mathematical way. User can change the preset value to change incoming damage from enemy champion, it means this tool will give different result base on battle condition.

Assumptions

The project is based on which item should we build for our hero to survival from an enemy hero's attack for 30 seconds. The attack speed is 1.3 hit per second and the physical damage is 160 per hit. The physical damage has a critical change to increase the damage into 1.2 times of the original damage. The magic damage is independent from attack speed and it is 80 magic damage per second.

Preset value List:

T=30 second

Attack speed =1.3 hit/sec

Critical chance = 30%

Physical Damage = 160 damage/hit

Magic damage = 80 damage/sec

Gold = 20000

Variables

The five defensive variables in Dota2 are Armor, magic resist, evasion, health Point, and Health regeneration. This project goal is spend limited gold to build most defensive build, and it turns into we need to decide how much gold we will spend on each variables.

Base on the game data, 2 armor is worth 175 gold, so we get:

Armor = Money. Armor*2/175 (prefer step size 175)

$A = x_1*2/175$

15% magic damage reduction is worth 550 gold then:

MR = Money.MR*0.15/550 (prefer step size 550)

$MR = x_2*0.15*550$

10% evasion chance is worth 1800 gold:

Evasion=Money. Evasion*0.1/1800(prefer step size 1800)

$E = x_3*0.1/1800$

25 health point is worth 100 gold:

Health Point = 25*Money.HP/100 (prefer step size 100)

$HP = 25*x_4/100$

1 health regeneration is worth 20 gold:

Health Regeneration = Money.HR*0.05 (prefer step size 20)

$HR = x_5*0.05$

Objective function

The objective function is the total health Point minus incoming damage:

Max HP = Total health – Total Damage

We can calculate the total health point by:

Health Point + Time*Health Regeneration

Then the income physical damage have to divide into two part, the first is non-critical hit damage:

- Time* Attack Speed* physical Damage* Damage multiplier *(1-critical chance)*(1-Evasion rate)

The second one is critical hit:

- Time* Attack Speed* physical Damage* Damage multiplier *critical chance*critical damage multiplier*(1-Evasion rate)

The Damage multiplier = 1 - 0.06 × armor ÷ (1 + (0.06 × |armor|)) is a nonlinear function.

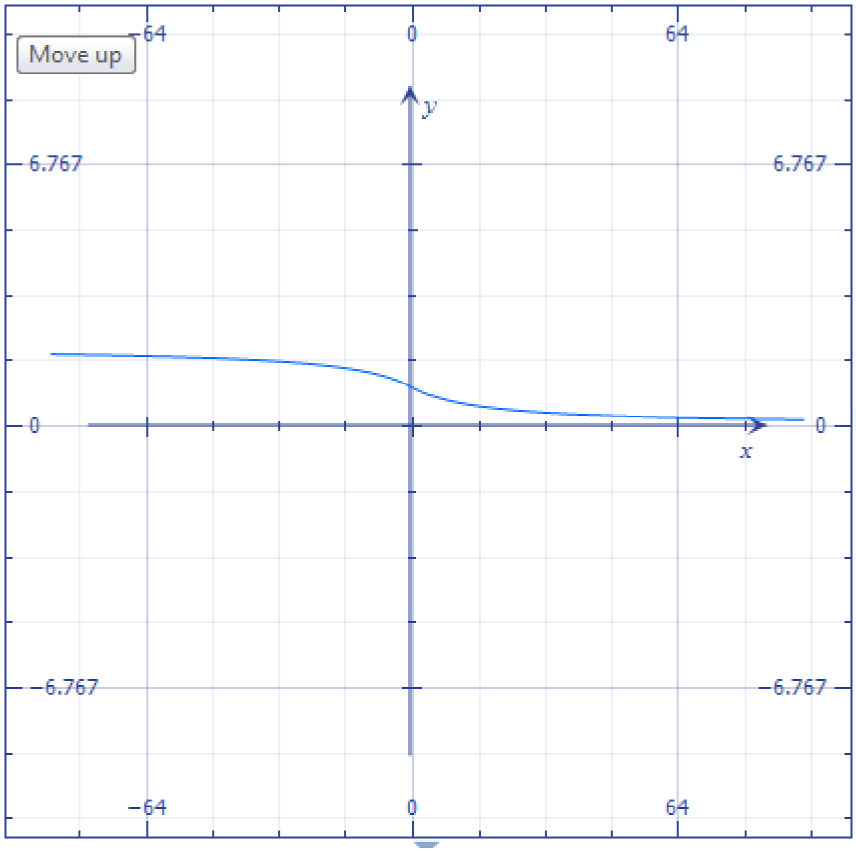


Figure 2: Damage reduction due to armor

And the total magic damage is calculate by:

- Time* Attack Speed* Magic damage * Magic Resist

The magic damage is not affect by critical hit and evasion rate which means hero cannot dodge the magic damage and magic damage cannot do critical hit.

And we substitute the Armor with x_1, Magic Resist with x_2, Evasion with x_3, Health Point with x_4, and Health Regeneration with x_5. The objective function turn into:

$$HP_{min}(x_1, x_2, x_3, x_4, x_5) = -(x_4*25/10 + x_5*0.05 - 30*1.3*160*0.7*(1-x_3*0.1/1800)*(1-0.06*(x_1*2/175)/(1+(0.06*(x_1*2/175))))-30*1.3*160*0.3*1.2*(1-x_3*0.1/1800)*(1-0.06*(x_1*2/175)/(1+(0.06*(x_1*2/175))))-30*80*(1-0.15*x_2/550))$$

Constraints

The project only have a one constrain money. Player can only have limited gold then:

Total cost <= Money spend on Armor + Money spend on Magic Resist + Money spend on Evasion + Money spend on Health Point + Money spend on Health Regenerate

So we get: g = x1+x2+x3+x4+x5-20000 <= 0

Design Optimization of a Table

By Devpreet Bhullar

I formulated an optimization problem for minimizing the material volume for a table. Buckling and bending stress provide with the two most important design constraints, while other constraints keep the table dimensions realistic and ergonomic for use by an average person. A distributed load of ~500 kg is assumed on the tabletop. The tabletop is rectangular and the cross-section of the table leg is square. The table has four legs, end connections for the legs are assumed to be fixed-free.

Maximum moment on the tabletop can be calculated (using $F = 500\ kg$) as:

$$M_{max} = R_1 \frac{L_t}{2} = \frac{F\left(\frac{L_t}{2}\right)^2}{L_t} = 125L_t\ [1]$$

$$R_1 = Reaction\ force\ on\ the\ tabletop, L_t = Length\ of\ the\ tabletop, F = load\ on\ the\ table$$

Section modulus is

$$S = \frac{bh^2}{6}\ [1]$$

$$b = Breadth\ of\ the\ tabletop, h = height\ of\ the\ tabletop$$

Constraint to make sure maximum bending stress for the tabletop is not reached:

$$\sigma_d bh^2 - 750L_t \leq 0\ [1] \Rightarrow \frac{750L_t}{\sigma_d bh^2} - 1 \leq 0$$

$$\sigma_d = Design\ stress\ for\ material$$

Constraint to ensure that Euler's formula can be used for buckling (column is long):

$$\sqrt{\frac{2\pi^2 E}{s_y}} - \frac{KL_l}{r} \leq 0\ [1]$$

$E = Modulus\ of\ elasticity\ of\ the\ material, s_y = Yield\ strength\ of\ the\ material, K = constant\ dependent\ of\ end\ fixity, L_l = Length\ of\ the\ table\ leg, r = radius\ of\ gyration$

Constraint to ensure table's leg doesn't buckle:

$$\frac{F}{2}-\frac{\pi^2 E a^2}{\frac{KL_l}{r}} \leq 0\ [1] \Rightarrow \frac{F}{2}-\frac{\pi^2 E a^2}{\frac{KL_l}{a/\sqrt{12}}} \leq 0$$

$$a = side\ of\ the\ table\ leg$$

Objective function

Let $x_1 = a, x_2 = L_l, x_3 = b, x_4 = h, x_5 = L_t$

Then the objective function can be written as: $minimize\ f(x_1, x_2, x_3, x_4, x_5) = 4x_1^2 x_2 + x_3 x_4 x_5$

Constraints

Subject to constraints

$$\frac{750x_5}{\sigma_d x_3 x_4^2} - 1 \leq 0$$

$$\sqrt{\frac{2\pi^2 E}{s_y}} - \frac{Kx_2}{x_1/\sqrt{12}} \leq 0$$

$$\frac{F}{2} - \frac{\pi^2 E x_1^2}{\frac{Kx_2}{x_1/\sqrt{12}}} \leq 0$$

$$x_3 - x_5 \leq 0, -x_1 + 0.01 \leq 0, -x_2 + 0.5 \leq 0, -x_3 + 0.3 \leq 0, -x_4 + 0.001 \leq 0, -x_5 + 0.6 \leq 0$$

$$x_1 - 0.2 \leq 0, x_2 - 0.6 \leq 0, x_3 - 0.4 \leq 0, x_4 - 0.1 \leq 0, x_5 - 1 \leq 0$$

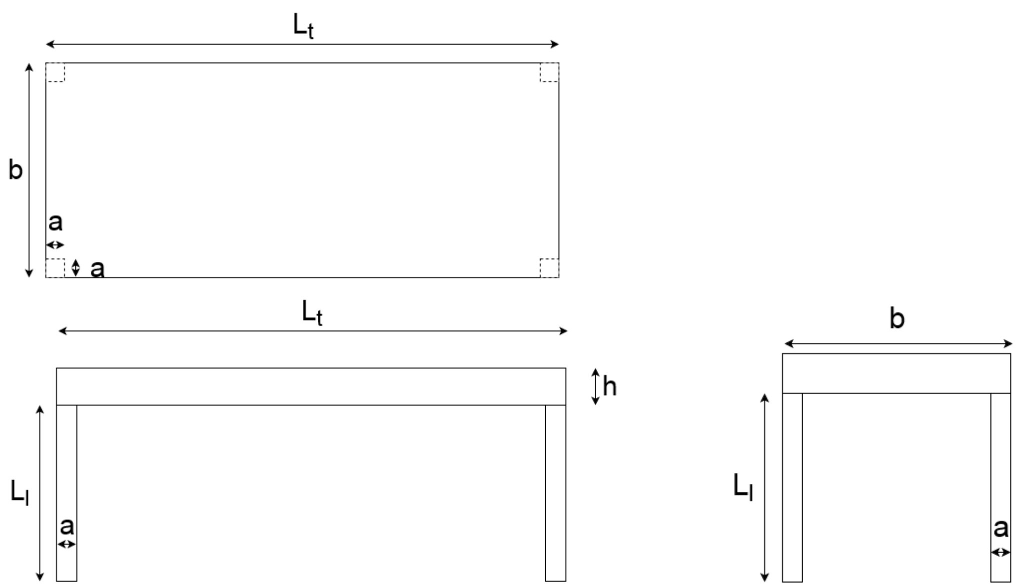


Figure 3: Counter-Clockwise from the top: a) Top view of the table. 2) Front view of the table. 3) Side view of the table.

Material for the table is chosen to be Douglas fir wood. Parameters for the material are as follows:

$$\sigma_d = 11.5\ MPa\ [2]$$

$$E = 13\ GPa\ [3]$$

$$s_y = 47.9\ MPa\ [4]$$

The following parameters are also known

$$K = 2.1\ [1]$$

$$F = 5000\ N$$

Method Selection

Since the problem is constrained and nonlinear in nature, we began by using *fmincon* to see if the problem involves more than one local minimum and also get a general idea of the optimization problem. We chose to use the default algorithm ('interior-point') unless faced by any errors. Although *fmincon* is greatly dependent on the starting point, choosing different starting points and analyzing the *fval* can give us an understanding of the nature of the function.

After using *fmincon*, we can use Genetic Algorithm to confirm the results given by *fmincon.* GA is a global minimizer, therefore if results from GA and *fmincon* match then we can deduce that the local minimum found by *fmincon* is indeed the global minimum, in which case the results provide us with the optimal point. If the results do not match between GA and *fmincon* or a sound conclusion cannot be made after using both methods, another method such as OASIS can be used to reach a conclusion about the optimization problem.

Solution Report

The objective function was firstly minimized using MATLAB's '*fmincon*' and was subjected to the constraints mentioned earlier. The initial values and their corresponding results are displayed below.

Table 1: Table outlining fmincon results for different starting points

Iteration	x_0	$x_1(a)$	$x_2(L_l)$	$x_3(b)$	$x_4(h)$	$x_5(L_t)$	$f(x)$
1	[0.01,0.3,0.2,0.05,0.5]	0.0409	0.5559	0.3469	0.0501	0.8028	0.0177
2	[0.20,0.6,0.4,0.1,1]	0.0400	0.5508	0.3471	0.0428	0.8194	0.0157
3	[0.01,0.5,0.3,0.001,0.6]	0.0100	0.5008	0.3001	0.0114	0.6001	0.0023

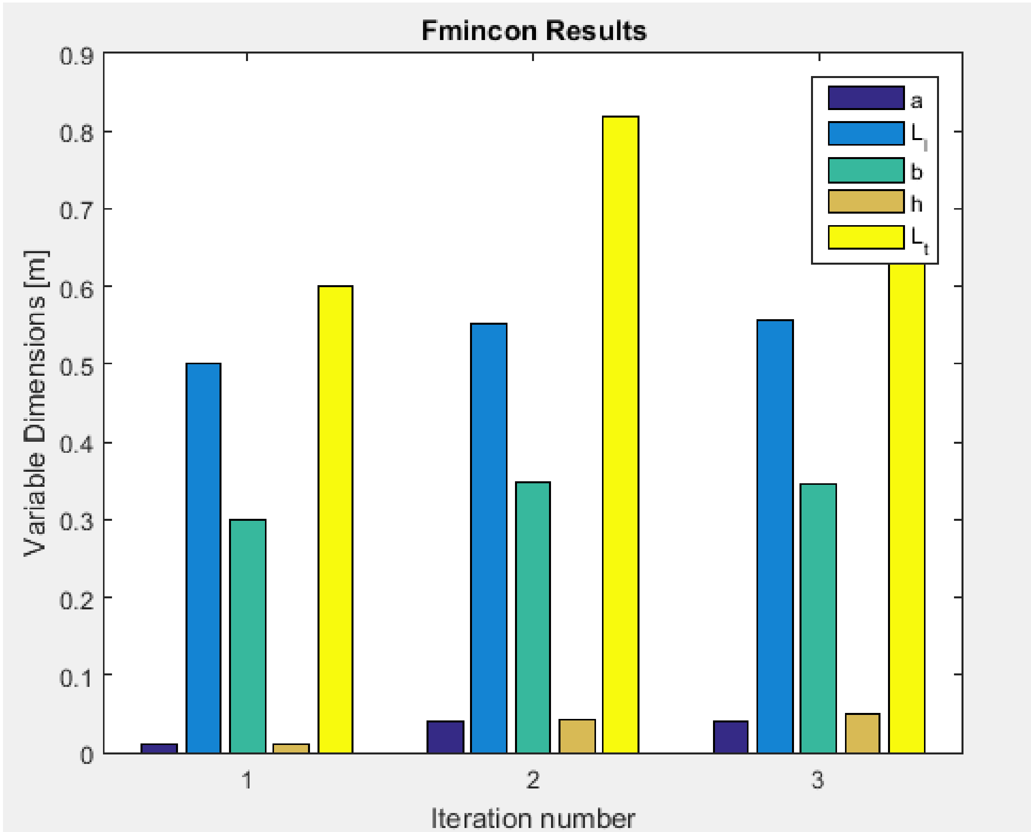


Figure 4: Bar chart comparing optimized variables values produced from fmincon

After using ‘*fmincon’,* we used MATLAB’s *‘ga’* toolbox to optimize the objective function again to check our solution from before. The minimization results produced by the genetic algorithm are shown below:

Table 2: Table outlining GA results for different iterations

Iteration/variable	$x_1(a)$	$x_2(L_l)$	$x_3(b)$	$x_4(h)$	$x_5(L_t)$	$f(x)$
1	0.0100	0.6000	0.3569	0.0114	0.7109	0.0031
2	0.0100	0.5766	0.3000	0.0184	0.6000	0.0035
3	0.0100	0.5971	0.3000	0.0166	0.6000	0.0032
4	0.0100	0.5999	0.3134	0.0120	0.6054	0.0025
5	0.0102	0.5998	0.3006	0.0166	0.6002	0.0032

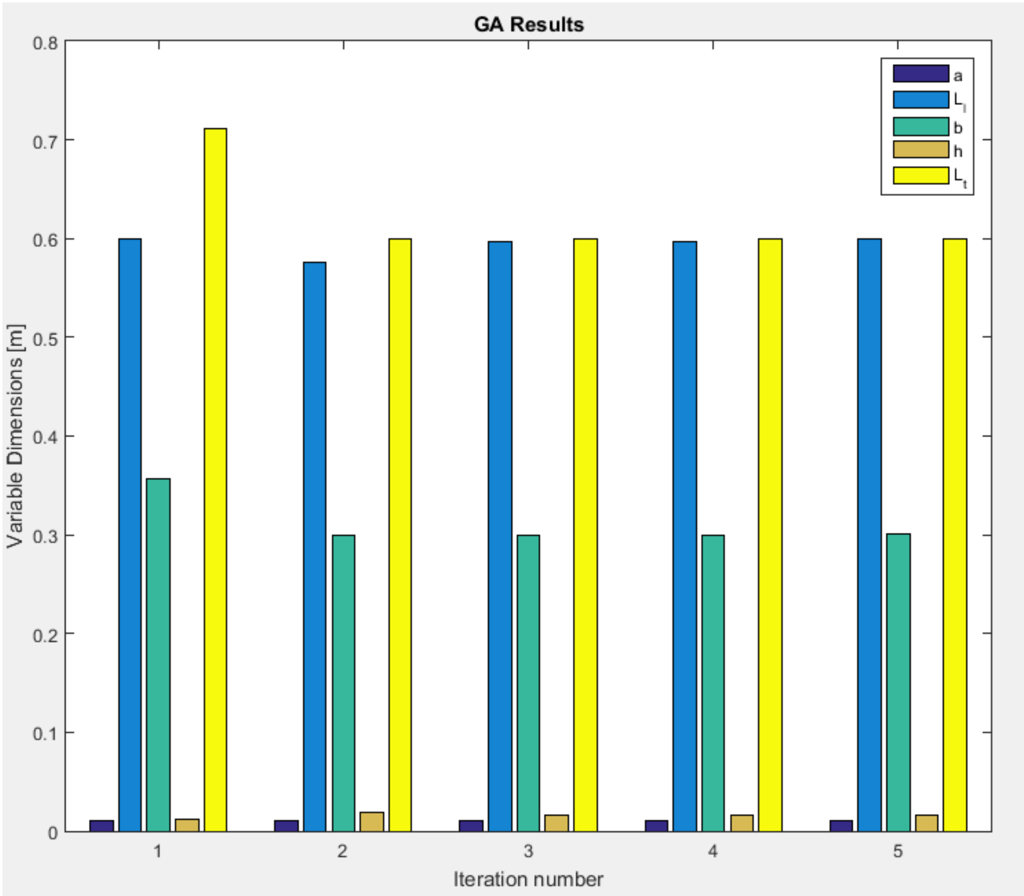


Figure 5: Bar chart comparing optimized variables values produced from GA

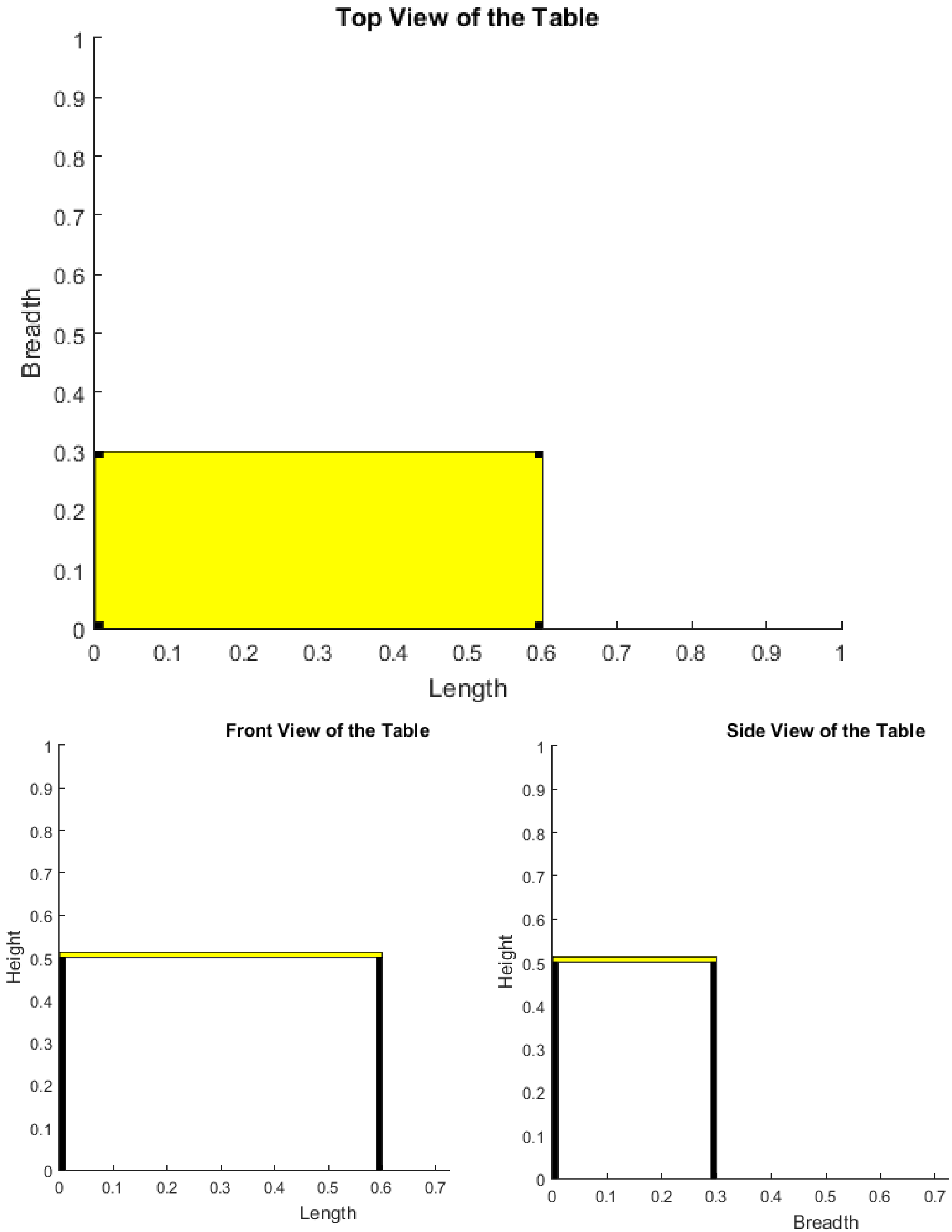


Figure 6: Dimensions obtained using fmincon: Counter-Clockwise from the top: a) Top view of the table. 2) Front view of the table. 3) Side view of the table.

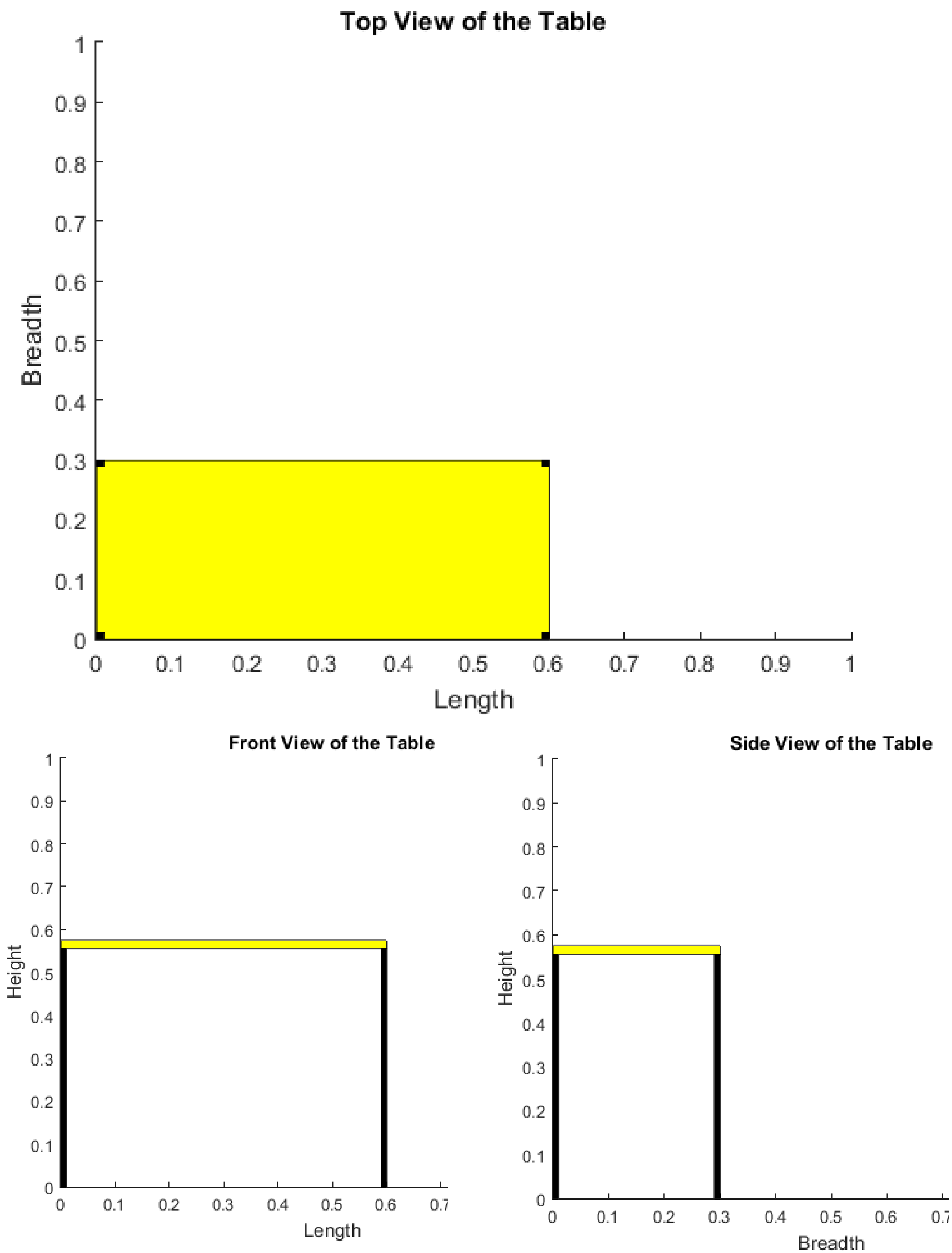


Figure 7: Dimensions obtained using GA: Counter-Clockwise from the top: a) Top view of the table. 2) Front view of the table. 3) Side view of the table.

Analysis

We compared the results to see if they are consistent. For *fmincon,* we used 80,000 maximum function evaluation limit and 10,000 as maximum iteration limit. The function was evaluated using different starting points as mentioned on the earlier section. Each iteration was run multiple times and exhibited the same result. A minimalized volume of 0.0023$m^3$was achieved when the lower bound was used as the starting point. This is very low compared to the results found with using other starting points.

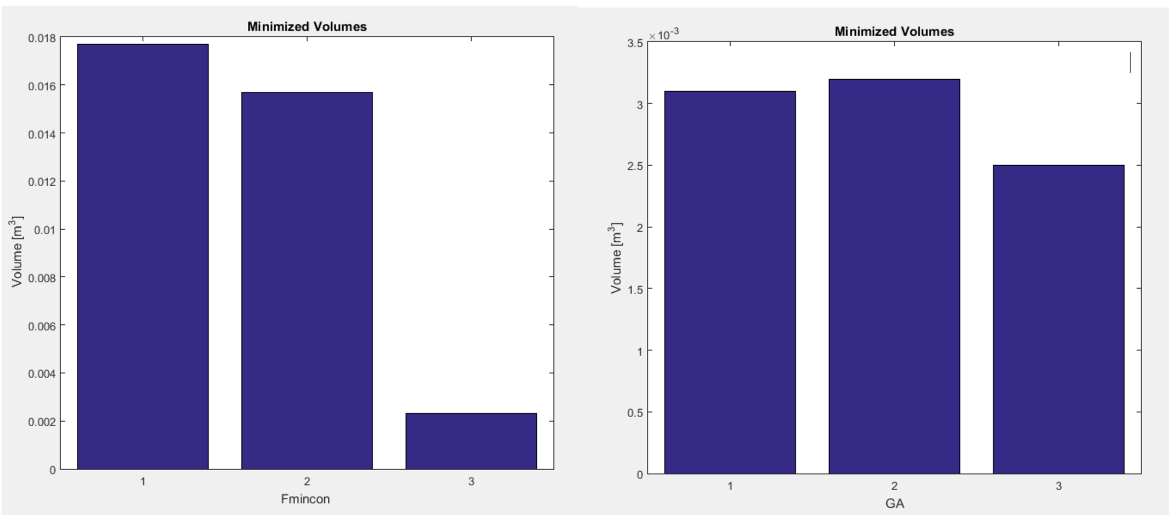


Figure 8: LEFT: Minimized volume comparison for different fmincons. RIGHT: Minimized volume comparison for different GA solutions

To double check the results from *fmincon*, we used GA to optimize the objective function. To make the search versatile, we used the '*mutationadaptfeaible*' function from the '*gaoptimset*' function. We used a population size of 2000 and used 10,000 generations to find the optimal solution of a minimalized volume of 0.0025m^3. All the minimized results from GA were very close to the minimum result found from *fmincon*. This further proves our hypothesis of the *fmincon* function being able to find the global optimum with lower bound as the starting points.

Table 3: bady

Solver	$x_1(a)$	$x_2(L_l)$	$x_3(b)$	$x_4(h)$	$x_5(L_t)$	$f(x)$
Fmincon	0.0100	0.5008	0.3001	0.0114	0.6001	0.0023
GA	0.0100	0.5999	0.3134	0.0120	0.6054	0.0025

Comparing the results from *fmincon* and GA puts the different optimized volumes into perspective. It shows how with different starting points, the *fmincon* solver was stuck at a local minimum around that starting point. The GA solver, however, was always able to get out of the local minimum and locate the global minimum. It could not pinpoint the minimum as it was limited by the number of population and generations. This was acceptable as the primary purpose of the GA was to inform us about the whereabouts of the global minimum to find an appropriate starting point for *fmincon.*

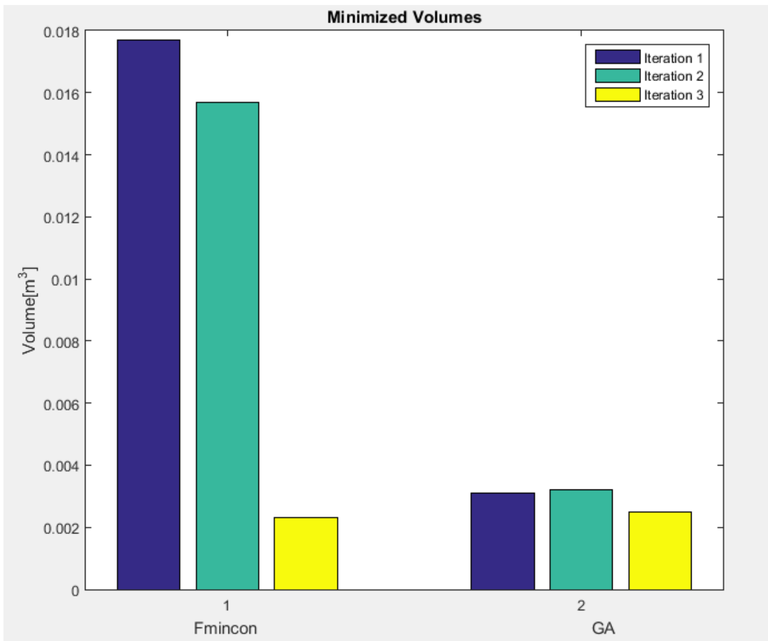


Figure 9: Minimized Volume comparison between fmincon and GA

Conclusion

The results from GA confirm the results from *fmincon* if the lower bound of variables is used as the starting point. However, *fval* obtained using *fmincon* was slightly better than that obtained by GA. We believe that GA could have been better if we used more iterations for GA, but this would have been time consuming which is why we didn't attempt it. Also we realize that GA, as a secondary tool, does confirm the results given by *fmincon* therefore it would be futile to use more iterations as the difference between the fval values can be considered negligible (0.0023 and 0.0025). Taking everything into account, we can conclude that we reached our goal in designing a four-legged Douglas fir wood table that can withhold about 500 kg of distributed load.

This project gave us an opportunity to apply our knowledge learned in the labs to a real-world problem chosen by us. This has helped us immensely in gaining a deeper understanding of optimization problems and tools available to solve such problems.

References

[1] R. L. Mott, Machine Elements in Mechanical Design, New Jersey: Pearson Education Inc., 2004.

[2] The Engineering Toolbox, "Wood Beams - Strength of Material," [Online]. Available: http://www.engineeringtoolbox.com/wood-beams-strength-d_1480.html. [Accessed March 2016].

[3] The Engineering Toolbox, "Young's Modulus - Tensile Modulus or Modulus of Elasticity - for some common materials," [Online]. Available: http://www.engineeringtoolbox.com/young-modulus-d_417.html. [Accessed March 2016].

[4] The Wood Database, "Douglar-fir- The Wood Database," [Online]. Available: http://www.wood-database.com/lumber-identification/softwoods/douglas-fir/.

Appendix

MATLAB codes

Main file:

```
clc;clear all;
A = [0,0,1,0,-1];
B = [0];
lb = [0.010,0.5,0.3,0.001,0.6];
ub = [0.20,0.6,0.4,0.1,1];

x0 = lb
% x0 = [0.001,0.001,0.001,0.001,0.001];
options.MaxFunEvals = 80000;
options.MaxIter = 10000;
[xflb,fvalf] = fmincon(@tableObjFun,x0,A,B,[],[],lb,ub,@confuneq,options)


x0 = ub
% x0 = [0.001,0.001,0.001,0.001,0.001];
options.MaxFunEvals = 80000;
options.MaxIter = 10000;
[xfub,fvalf] = fmincon(@tableObjFun,x0,A,B,[],[],lb,ub,@confuneq,options)


x0 = ub/2
x0 = [0.001,0.001,0.001,0.001,0.001];
options.MaxFunEvals = 80000;
options.MaxIter = 10000;
[xfub2,fvalf] = fmincon(@tableObjFun,x0,A,B,[],[],lb,ub,@confuneq,options)
```

```
options = gaoptimset;
options=gaoptimset(options,'populationsize',200,'Generations',4000,'Selection
Fcn',@selectionroulette)
options = gaoptimset(options,'MutationFcn',{@mutationadaptfeasible 0.01});
options = gaoptimset(options,'PlotFcns', {@gaplotbestf,@gaplotbestindiv})
[xg,fvalg,exitflag,output] =
ga(@tableObjFun,5,A,B,[],[],lb,ub,@confuneq,options);


% x =
[0.01,0.6,0.3569,0.0114,0.7109;0.01,.5766,.3,.0184,.6;0.01,.5971,.3,.0166,.6;
0.01,.5971,.3,.0166,.6,;0.0102,0.5998,.3006,.0166,.6002]
% y =[0.0177,0.0157,0.0023; 0.0031,0.0032,0.0025]
% bar(y)
% legend('Iteration 1','Iteration 2','Iteration 3');
% title('Minimized Volumes')
% ylabel('Volume[m^3]')
% xlabel('Fmincon, GA')
```

Constraints Function File

```
function [c,ceq] = confuneq(x)
%Parameters
E = 13*10^9;
sd = 11.5*10^6;
F = 5000;
sy = 47.9*10^6;
K = 2.1;
%Nonlinear inequality constraints
c = [sqrt(2*pi^2*E/sy)-K*x(2)/(x(1)/sqrt(12));
    F/4-pi^2*E*x(1)^2/(K*x(2)/(x(1)/sqrt(12)));
    ((750*x(5))/(sd*x(3)*x(4)^2))-1];
%Nonlinear equality constraints
ceq = [];
```

Objective Function File

```
function f = tableObjFun(x)
f = 4*x(1)^2*x(2)+x(3)*x(4)*x(5);
```