Design of Full Car Suspension Model to Maximize Rider Comfort

AE755 Course Project

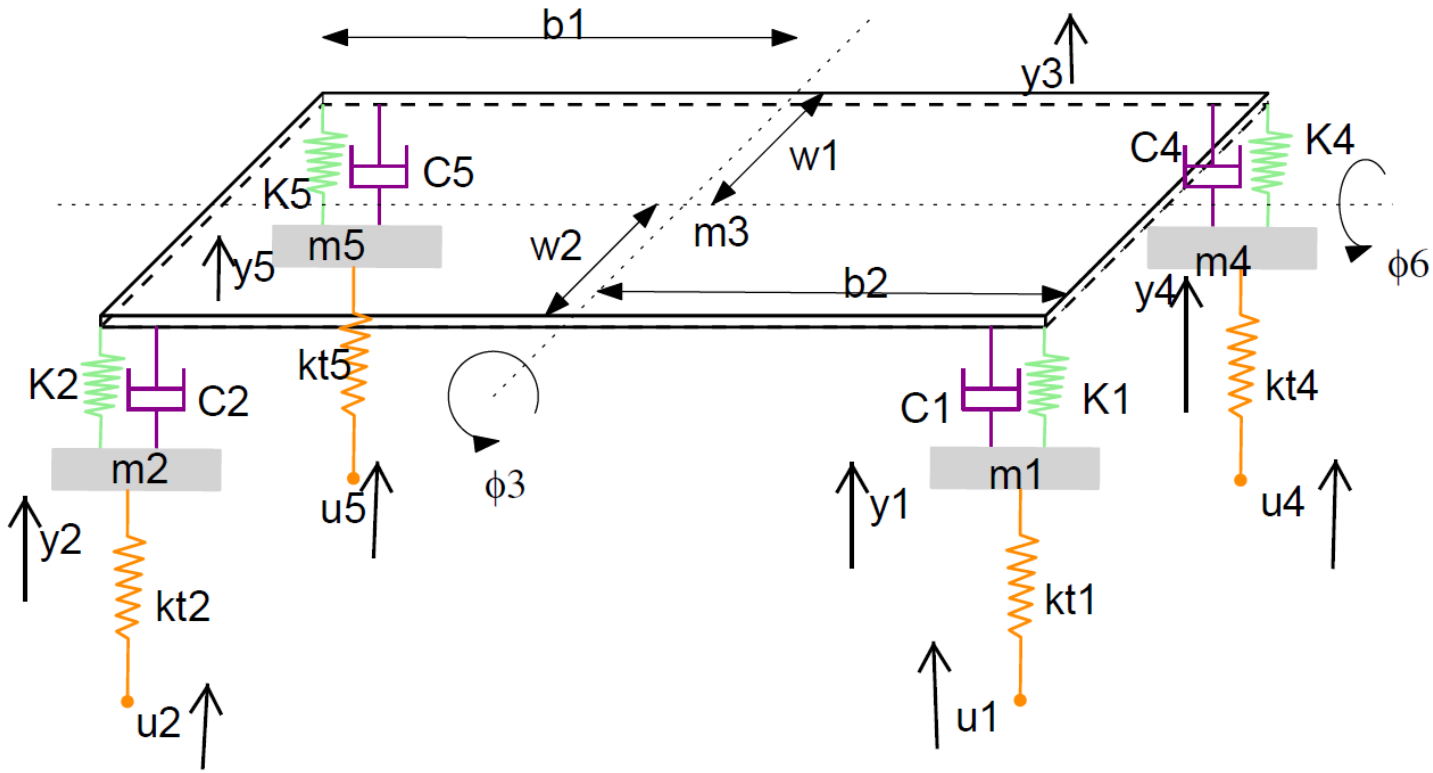
Team: Cars

(Under the guidance of Prof. Abhijit Gogulapati)

April 2021

Case: Level Road

- Assumption: u1 = u2 = u4 = u5 = 1



Equations of Motion

$$M\ddot{x} + C\dot{x} + Kx = 0$$

```
M =

[0, 0,  0, 0, 0,                                0,                                0]
[0, 0,  0, 0, 0,                                0,                                0]
[0, 0, m3, 0, 0,                                0,                                0]
[0, 0,  0, 0, 0,                                0,                                0]
[0, 0,  0, 0, 0,                                0,                                0]
[0, 0,  0, 0, 0, (m3*(b1^2 - b1*b2 + b2^2))/3,                                0]
[0, 0,  0, 0, 0,                                0, (m3*(w1^2 - w1*w2 + w2^2))/3]
```

Equations of Motion

```
Cref =

[  -C1,      0,                          C1,      0,      0,                                  C1*b1,                                  -C1*w2]
[    0,    -C2,                          C2,      0,      0,                                 -C2*b2,                                  -C2*w2]
[  -C1,    -C2,            C1 + C2 + C4 + C5,    -C4,    -C5,           C1*b1 - C2*b2 + C4*b1 - C5*b2,           C4*w1 - C2*w2 - C1*w2 + C5*w1]
[    0,      0,                          C4,    -C4,      0,                                  C4*b1,                                  C4*w1]
[    0,      0,                          C5,      0,    -C5,                                 -C5*w2,                                  C5*w1]
[C1*b1, -C2*b2, C1*b1 - C2*b2 + C4*b1 - C5*b2, -C4*b1,  C5*w2,     C1*b1^2 + C2*b2^2 + C4*b1^2 + C5*b2^2, C2*b2*w2 - C1*b1*w2 + C4*b1*w1 - C5*b2*w1]
[C1*w2,  C2*w2, C4*w1 - C2*w2 - C1*w2 + C5*w1, -C4*w1, -C5*w1, C2*b2*w2 - C1*b1*w2 + C4*b1*w1 - C5*b2*w1,     C1*w2^2 + C2*w2^2 + C4*w1^2 + C5*w1^2]
```

Equations of Motion

```
Kref =

[  -K1,      0,                            K1,      0,      0,                                    K1*b1,                                    -K1*w2]
[    0,    -K2,                            K2,      0,      0,                                   -K2*b2,                                    -K2*w2]
[  -K1,    -K2,             K1 + K2 + K4 + K5,    -K4,    -K5,             K1*b1 - K2*b2 + K4*b1 - K5*b2,             K4*w1 - K2*w2 - K1*w2 + K5*w1]
[    0,      0,                            K4,    -K4,      0,                                    K4*b1,                                     K4*w1]
[    0,      0,                            K5,      0,    -K5,                                   -K5*w2,                                     K5*w1]
[K1*b1, -K2*b2, K1*b1 - K2*b2 + K4*b1 - K5*b2, -K4*b1,  K5*w2,     K1*b1^2 + K2*b2^2 + K4*b1^2 + K5*b2^2, K2*b2*w2 - K1*b1*w2 + K4*b1*w1 - K5*b2*w1]
[K1*w2,  K2*w2, K4*w1 - K2*w2 - K1*w2 + K5*w1, -K4*w1, -K5*w1, K2*b2*w2 - K1*b1*w2 + K4*b1*w1 - K5*b2*w1,     K1*w2^2 + K2*w2^2 + K4*w1^2 + K5*w1^2]
```

Preprocessing for the Decoupled equations

```
>> pretty(Y3(j*w))

/
|
|                                                                    #10 #1
| K1 #16 + K2 #15 + K4 #14 + K5 #13 - C1 #9 - C2 #8 - C4 #7 - C5 #6 - ------
\                                                                      #2

        / #12 #3                                                 #4 #1                                                        \ \
    #12 | ------ + C1 b1 #9 - C2 b2 #8 + C4 b1 #7 - C5 b2 #6 + ----- - K1 b1 #16 + K2 b2 #15 - K4 b1 #14 + K5 b2 #13 | |
        \   #11                                                   #2                                                          / |
  + ------------------------------------------------------------------------------------------------------------------------ |/#11
                                                       #5                                                                     /

where

        / #12 #3                                                                                                    \
    #4 | ------ + C1 b1 #9 - C2 b2 #8 + C4 b1 #7 - C5 b2 #6 - K1 b1 #16 + K2 b2 #15 - K4 b1 #14 + K5 b2 #13 |
        \   #11                                                                                                     /
   #1 == ----------------------------------------------------------------------------------------------------------------
                                                       #5

                                                                                                      #10 #3
    - C1 w2 #9 - C2 w2 #8 + C4 w1 #7 + C5 w1 #6 + K1 w2 #16 + K2 w2 #15 - K4 w1 #14 - K5 w1 #13 - ------
                                                                                                         #11

                                                                  2     2                                             2    2           2
             2          2        2        2        2      #4    #10          2        2        2    m3 w  (w1  - w1 w2 + w2 )
   #2 == K1 w2  + w (C1 w2  + C2 w2  + C4 w1  + C5 w1 ) 1i - --- - ---- + K2 w2  + K4 w1  + K5 w1  - ------------------------
                                                              #5    #11                                              3

   #3 == K1 #16 + K2 #15 + K4 #14 + K5 #13 - C1 #9 - C2 #8 - C4 #7 - C5 #6

                                                                                                          #10 #12
   #4 == w (C1 b1 w2 - C2 b2 w2 - C4 b1 w1 + C5 b2 w1) 1i + K1 b1 w2 - K2 b2 w2 - K4 b1 w1 + K5 b2 w1 - -------
                                                                                                            #11

                 2                                                                            2    2            2
            2   #12        2        2        2          2        2        2        2       m3 w  (b1  - b1 b2 + b2 )
```

Preprocessing for the Decoupled equations

$$\#4 == w\,(C1\,b1\,w2 - C2\,b2\,w2 - C4\,b1\,w1 + C5\,b2\,w1)\,1i + K1\,b1\,w2 - K2\,b2\,w2 - K4\,b1\,w1 + K5\,b2\,w1 - \frac{\#10\ \#12}{\#11}$$

$$\#5 == K1\,b1^2 - \frac{\#12^2}{\#11} + K2\,b2^2 + K4\,b1^2 + K5\,b2^2 + w\,(C1\,b1^2 + C2\,b2^2 + C4\,b1^2 + C5\,b2^2)\,1i - \frac{m3\,w^2\,(b1^2 - b1\,b2 + b2^2)}{3}$$

$$\#6 == u5(0) - w\ \#13\ 1i$$

$$\#7 == u4(0) - w\ \#14\ 1i$$

$$\#8 == u2(0) - w\ \#15\ 1i$$

$$\#9 == u1(0) - w\ \#16\ 1i$$

$$\#10 == K1\,w2 + K2\,w2 - K4\,w1 - K5\,w1 + w\,(C1\,w2 + C2\,w2 - C4\,w1 - C5\,w1)\,1i$$

$$\#11 == K1 + K2 + K4 + K5 - m3\,w^2 + w\,(C1 + C2 + C4 + C5)\,1i$$

$$\#12 == K1\,b1 - K2\,b2 + K4\,b1 - K5\,b2 + w\,(C1\,b1 - C2\,b2 + C4\,b1 - C5\,b2)\,1i$$

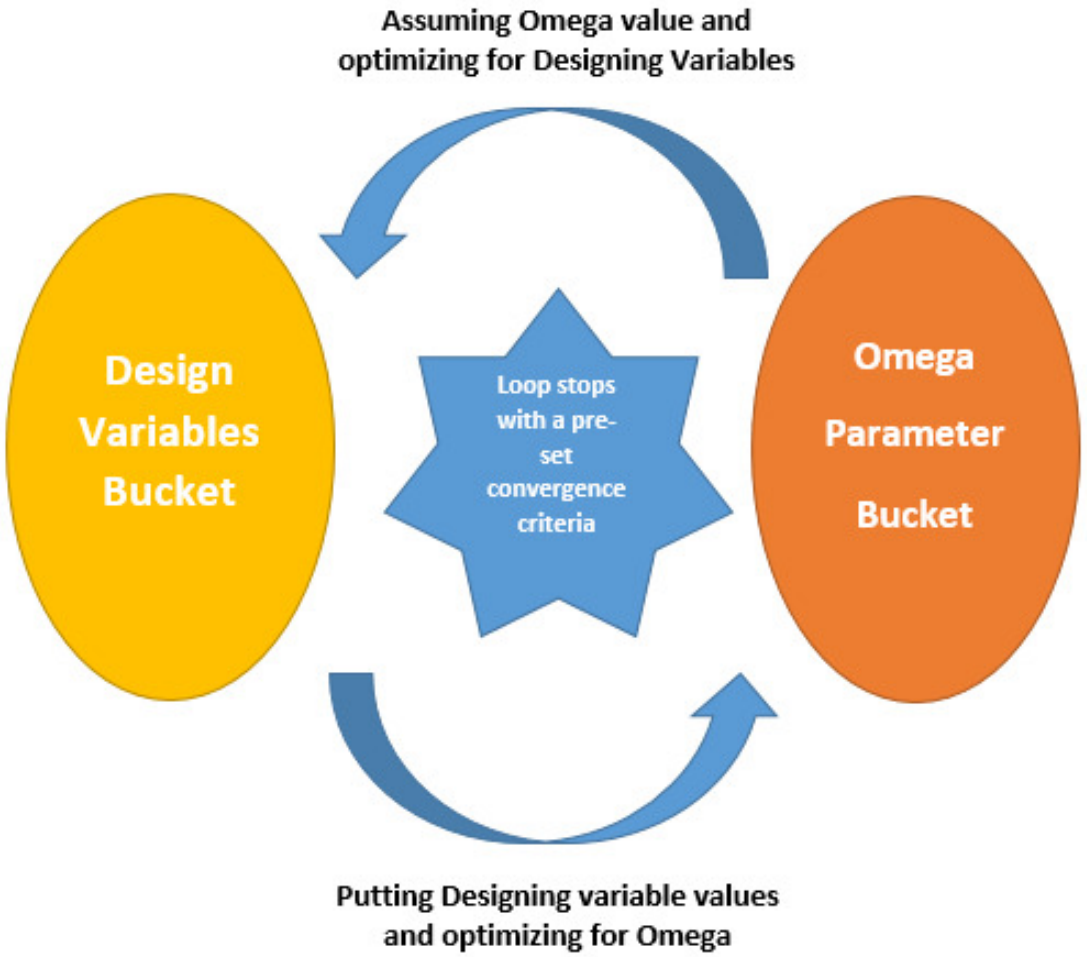
$$\#13 == \mathrm{laplace}(u5(t),\ t,\ w\ 1i)$$

$$\#14 == \mathrm{laplace}(u4(t),\ t,\ w\ 1i)$$

$$\#15 == \mathrm{laplace}(u2(t),\ t,\ w\ 1i)$$

$$\#16 == \mathrm{laplace}(u1(t),\ t,\ w\ 1i)$$

Tackling The Optimization Problem



Approach 1 (Benchmarking against Test functions)

Name of the function	**Number of Design variables**	**Global Minima**	**Local Minima**
$x^2 + (x-3)^2$	10	1.5	1.5
$(x-40) * (x-10) * (x+10) * (x+30)$	10	29.394	-22.237
Himmelblau's function	2	(3,2) (-2.80,3.13) (-3.77,-3.28) (3.58,-1.84)	(3,2) (-2.80,3.13) (-3.77,-3.28) (3.58,-1.84)
Banana function	2	(1,1)	(1,1)

Approach 2 for SA (Benchmarking against Built in functions)

Design Variable	Optimal value from Our algo	Optimal value from Built-in algo	Design Variable	Optimal value from Our algo	Optimal value from Built-in algo
k1	21656.5	30180	c5	674.62	640
k2	27165.88	30180	b1	2	3.5
k4	28177.37	30180	b2	3.5	2
k5	23394.45	30180	w1	0.75	0.75
c1	640	640	w2	1.2	1.2
c2	722.66	640	omega	100	-100
c4	670.45	640	Obj function	0.0036	0.0039

Approach 2 for PSO (Benchmarking against Built in functions)

Design Variable	Optimal value from Our algo	Optimal value from Built-in algo	Design Variable	Optimal value from Our algo	Optimal value from Built-in algo
k1	30180	30180	c5	640	640
k2	30180	30180	b1	3.5	2
k4	30180	30180	b2	2	2
k5	30180	30180	w1	1.2	0.75
c1	640	640	w2	1.2	0.75
c2	640	640	omega	-100	-100
c4	640	640	Obj function	0.0038	0.0038

Approach 2 for GA (Benchmarking against Built in functions)

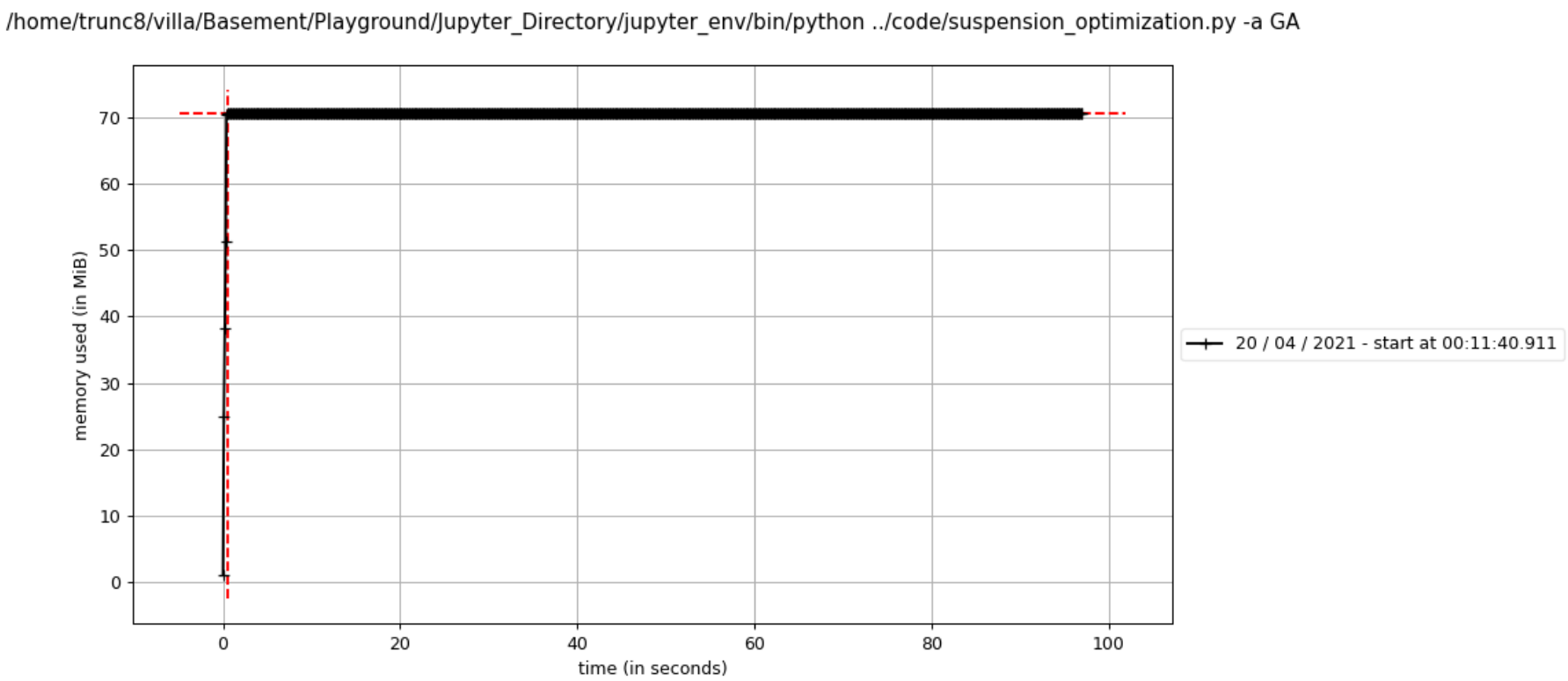
Design Variable	Optimal value from Our algo	Optimal value from Built-in algo	Design Variable	Optimal value from Our algo	Optimal value from Built-in algo
k1	22552.08	29925.1	c5	959.18	648.49
k2	20121.11	29905.8	b1	2.24	2.23
k4	20121.02	30011.4	b2	3.48	2.5
k5	30178.97	29525.2	w1	1.19	0.99
c1	956.77	643.48	w2	0.74	1.19
c2	956.63	644.14	omega	0	-100
c4	959.31	640.29	Obj function	14557.38	0.0038

Time Taken

- Time taken by GA: 96.579s
- Time taken by GA_benchmark: 10.603s
- Time taken by SA: 209s
- Time taken by SA_benchmark: 59.980s
- Time taken by PSO: 313.783s
- Time taken by PSO_benchmark: 879.916s

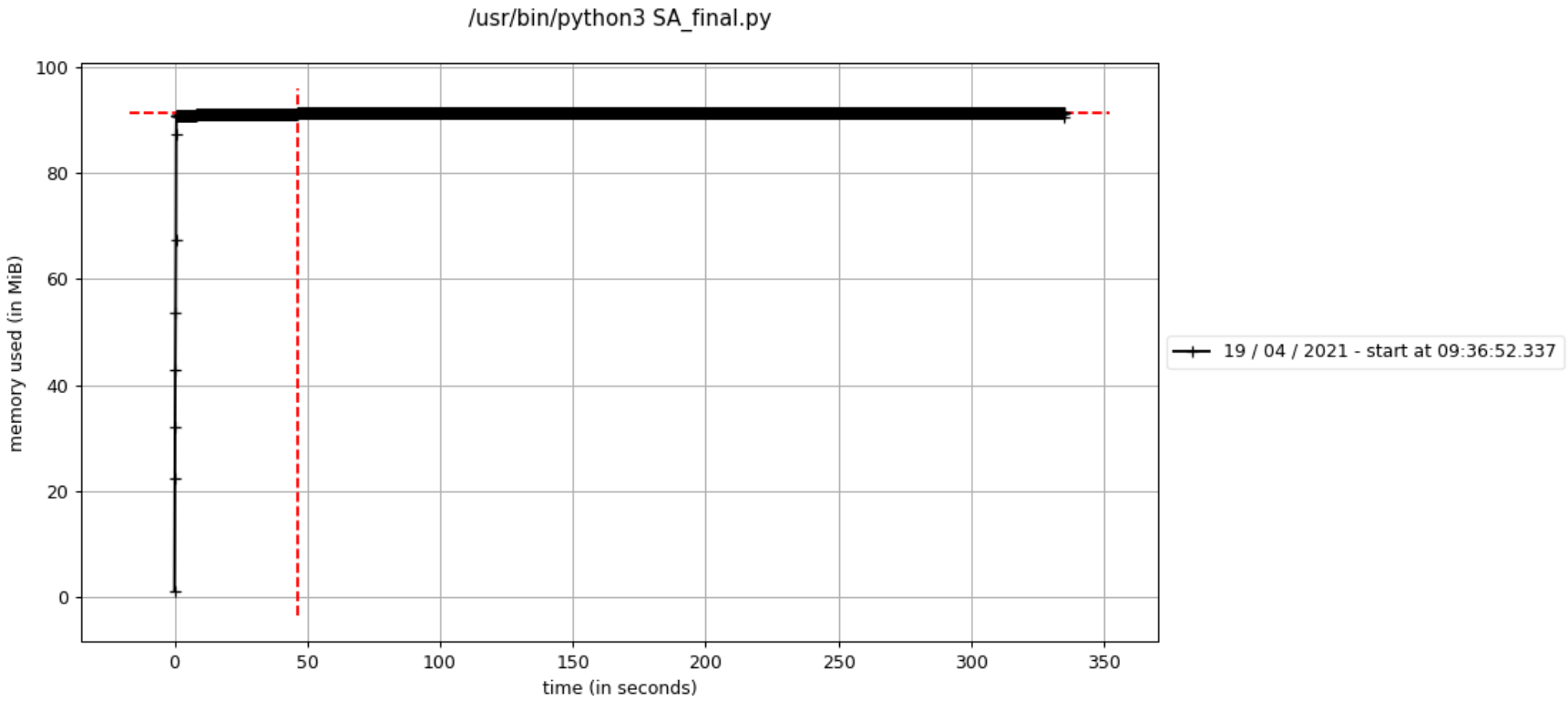
Computational Expenses (Algos written from scratch)

Genetic Algorithm:



Computational Expenses (Algos written from scratch)

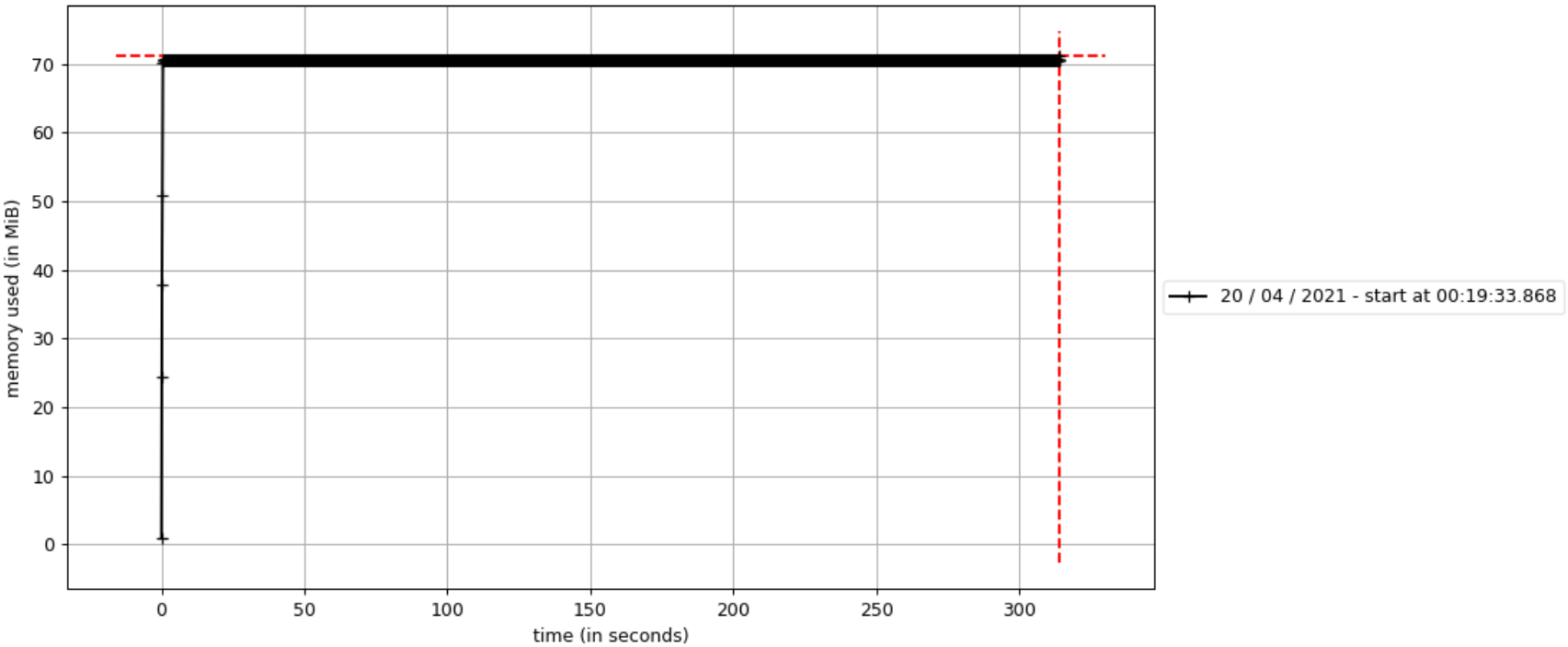
Simulated Annealing:



Computational Expenses (Algos written from scratch)

Particle Swarm Optimization:

/home/trunc8/villa/Basement/Playground/Jupyter_Directory/jupyter_env/bin/python ../code/suspension_optimization.py -a PSO

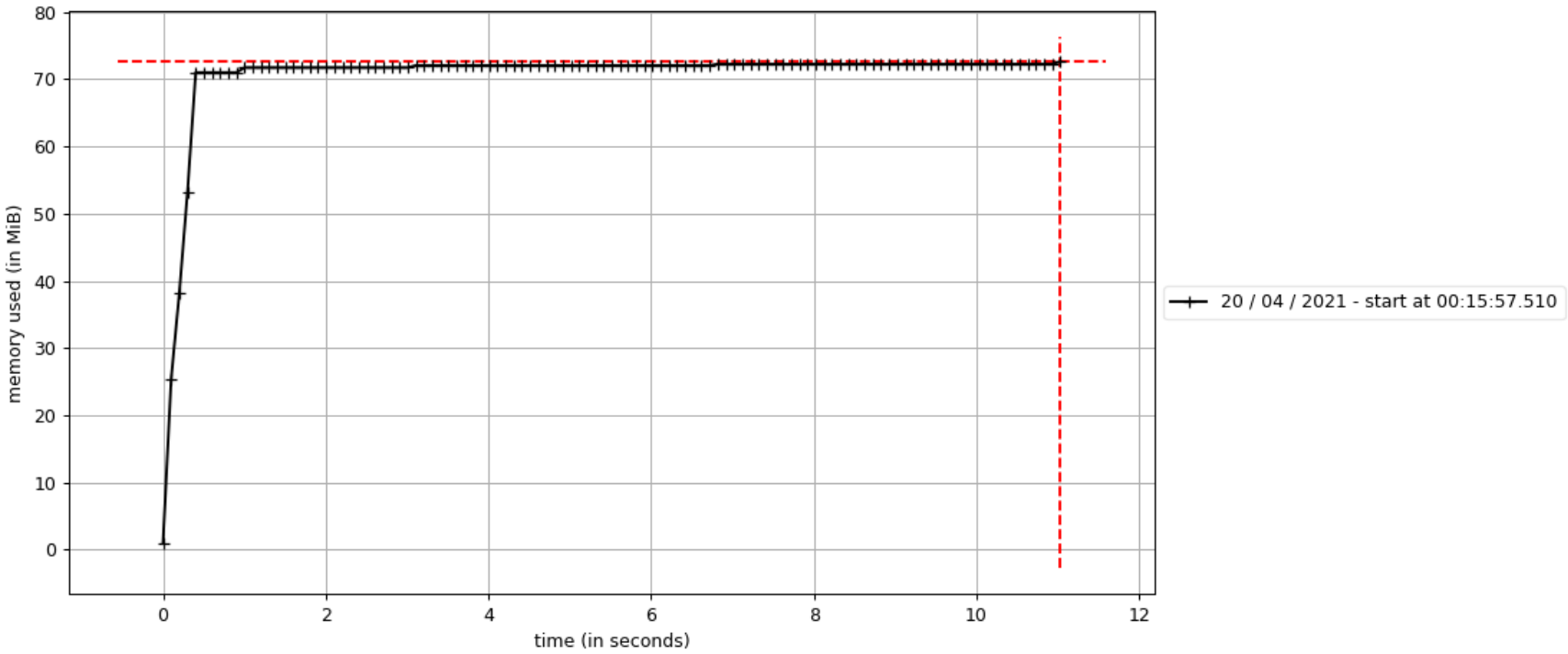


GA has the lowest MiB(70) and therefore it is the most Memory efficent

Computational Expenses (Built in algos)

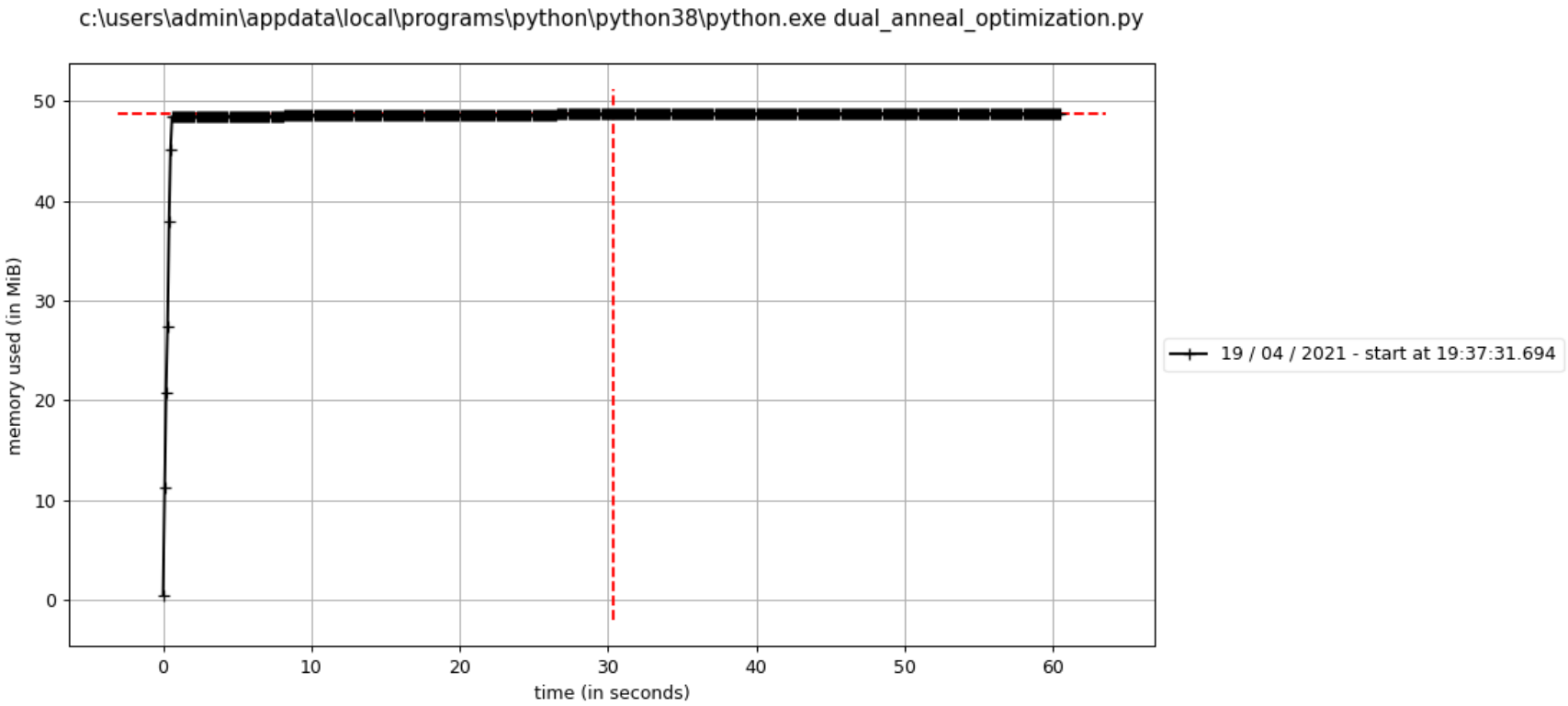
Genetic Algorithm:

ome/trunc8/villa/Basement/Playground/Jupyter_Directory/jupyter_env/bin/python ../code/suspension_optimization.py -a GA_benchmark



Computational Expenses (Built in algos)

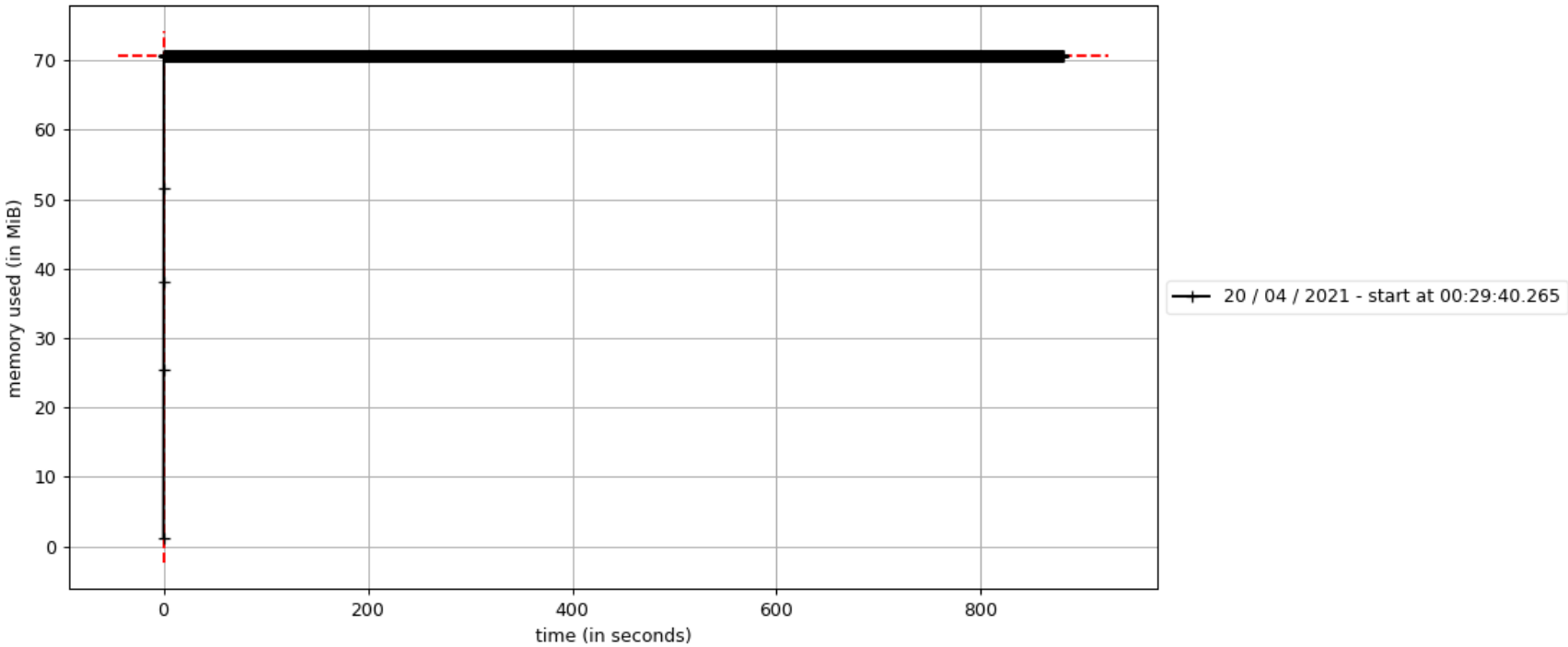
Simulated Annealing:



Computational Expenses (Built in algos)

Particle Swarm Optimization:

ɔme/trunc8/villa/Basement/Playground/Jupyter_Directory/jupyter_env/bin/python ../code/suspension_optimization.py -a PSO_benchmark



SA has the lowest MiB(48) and therefore it is the most Memory efficent

Comparing predictions of the optimal points of all the 3 algos

Design Variable	GA	SA	PSO	Design Variable	GA	SA	PSO
k1	22552.08	21656.5	30180	c5	959.18	674.62	640
k2	20121.11	27165.88	30180	b1	2.24	2	3.5
k4	20121.02	28177.37	30180	b2	3.48	3.5	2
k5	30178.97	23394.45	30180	w1	1.19	0.75	1.2
c1	956.77	640	640	w2	0.74	1.2	1.2
c2	956.63	722.66	640	omega	0	100	-100
c4	959.31	670.45	640	Obj function	14557.38	0.0036	0.0038

Comparing Sensitivity values of Design variables

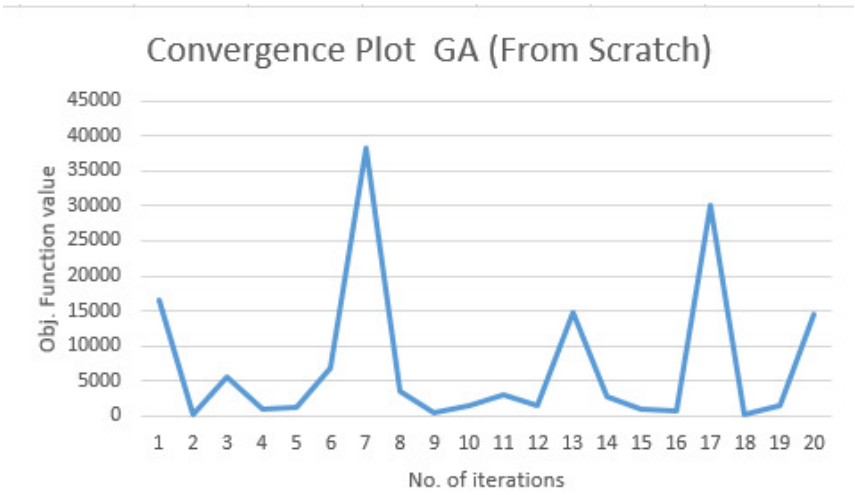
Design Variable	GA	GA bench-mark	SA	SA bench-mark	PSO	PSO bench-mark
k1	985442.61	1.36×10^{-5}	3.33×10^{-5}	9999.99	10000.00	10000.00
k2	985442.61	1.36×10^{-5}	3.33×10^{-5}	9999.99	10000.00	10000.00
k4	985442.61	1.36×10^{-5}	3.33×10^{-5}	9999.99	10000.00	10000.00
k5	985442.61	1.36×10^{-5}	3.33×10^{-5}	9999.99	10000.00	10000.00
c1	985442.61	1.46×10^{-5}	3.21×10^{-5}	4.97×10^{-5}	4.00×10^{-5}	3.37×10^{-5}
c2	985442.61	1.46×10^{-5}	3.16×10^{-5}	5.03×10^{-5}	4.00×10^{-5}	3.37×10^{-5}
c4	985442.61	1.46×10^{-5}	3.17×10^{-5}	5.03×10^{-5}	4.00×10^{-5}	3.37×10^{-5}

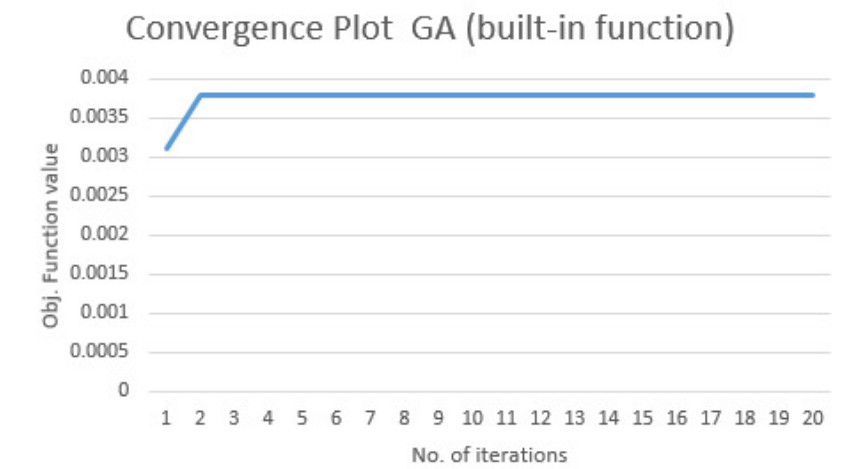
Comparing Sensitivity values of Design variables

Design Variable	GA	GA bench-mark	SA	SA bench-mark	PSO	PSO bench-mark
c5	985442.61	1.46×10^{-5}	3.22×10^{-5}	4.99×10^{-5}	4.00×10^{-5}	3.37×10^{-5}
b1	985442.61	1.21×10^{-5}	5.41×10^{-5}	9999.99	10000.00	3.84×10^{-5}
b2	985442.61	9.84×10^{-6}	10000.00	5.54×10^{-5}	3.53×10^{-5}	3.84×10^{-5}
w1	985442.61	6240.99	3025.00	3024.99	10000.00	3025.00
w2	985442.61	9801.00	9999.99	10000.00	10000.00	3025.00
omega	14555.37	5.47×10^{-5}	3.61×10^{-5}	2.04×10^{-5}	0.00011	0.00010

Convergence Plots

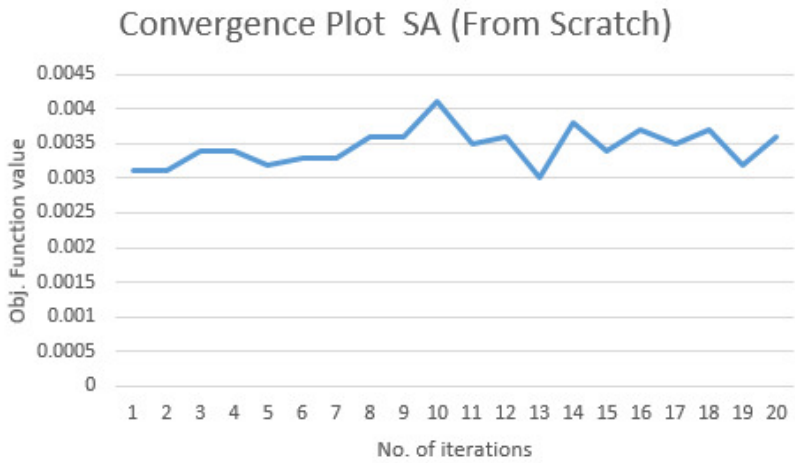
Genetic Algorithm:

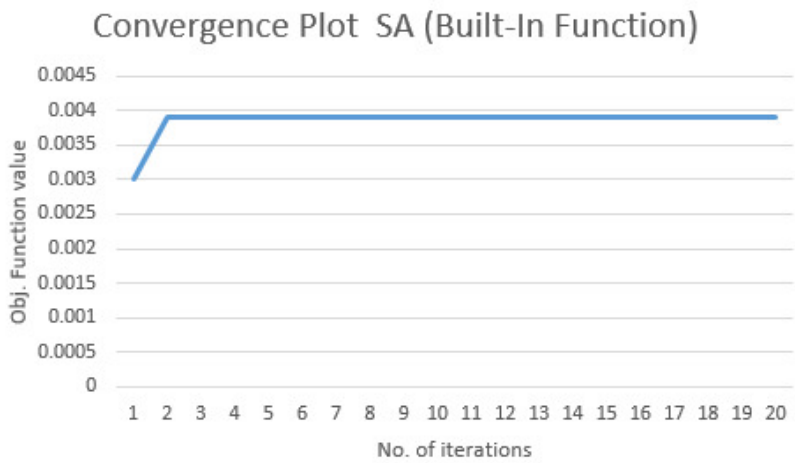




Convergence Plots

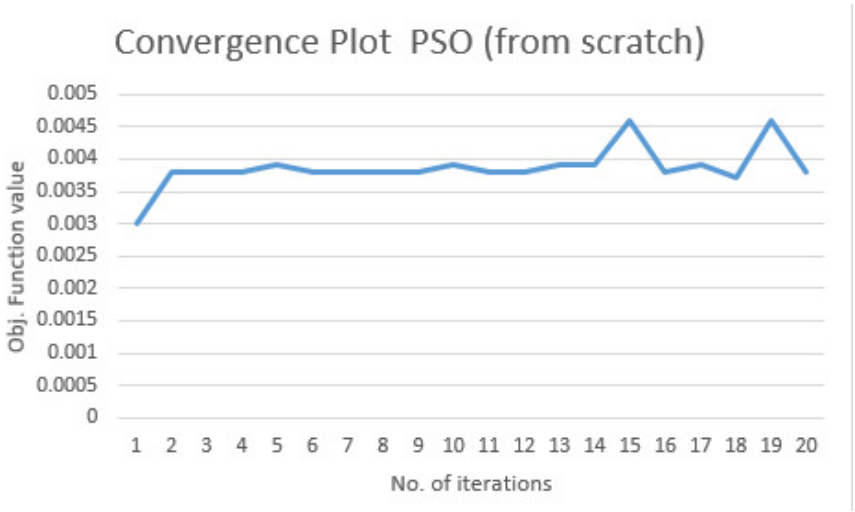
Simulated Annealing:

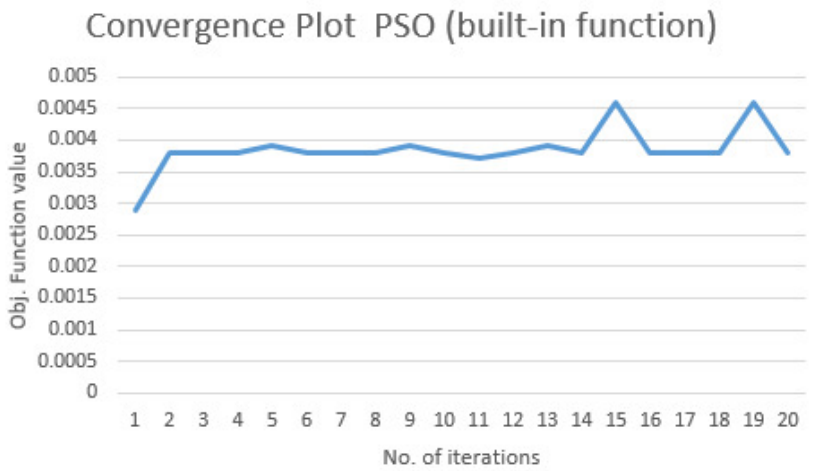




Convergence Plots

Particle Swarm Optimization:





Learning and Recommendations

- The SA algorithm written from scratch surpasses the other 2 algos in optmization of objective function
- The SA algo even produces better results than the Built in algo from SciPy library
- Only those design variables which have a higher impact on the objective function should be considered important and others will be safe to neglect.
- Based on the sensitivity report obtained for each algorithm we see:
  1. Variables k1,k2,k4,k5 have highest sensitivity
  2. Following them are the sensitivity values of b1,b2,w1,w2
  3. omega is the least one in terms of sensitivity
- Based on this, we say, the important design variable are:
  k1,k2,k4,k5 followed by b1,b2,w1,w2,c1,c2,c4,c5
  (All mentioned in descending order of their priority)

Our Recommendation

The final optimized parameters that we recommend are as follows:

- Stiffness (in $N.m^{-1}$): 21656.5, 27165.88, 28177.37, 23394.45
- Damping (in $N.s.m^{-1}$): 640, 722.66, 670.45, 674.62
- Center of mass distance from the ends of the car chassis
  - along length (in m): 2 and 3.5
  - along width (in m): 0.75 and 1.2

Team Details

- Siddharth Saha (170100025)
- Manthan Dhisale (193109014)
- Chinmay Gandhshreewar (193109018)
- Nabajyoti Majumdar (170100081)
- Saurabh Parekh (170100016)
- Rohith Janjanam (180050041)
- Ishan Phansalkar (19D070046)

References/Links

1. Genetic Algorithm
2. Simulated Annealing
3. Numerical Optimisation by Jorge Nocedal and Stephen J. Wright, Chapter 18
4. Acta Numerica, Sequential Quadratic Programming by Boggs and Tolle
5. GitHub repository of our project