Blast Off!

Lecture 9

Application of MOGA to SSTO rocket models

Quick Recap

We're using MOGA to increase payload mass (J_1) while simultaneously decreasing total cost (J_2)

Multiple individuals are randomly generated and the best among them reproduce to populate the next generation. But how exactly do we generate these random rocket models? Let's look into the pricing first..

Cost Calculation

1. Engine Cost

Engine mass and Engine cost can be said to have a proportional ratio. Hence, Space Shuttle can be used as a good reference.

$$C_E = M_E * (C_{Ess} / M_{Ess})$$

C_{Ess} = 30 million \$

M_{Ess} = 3,500 kg

Cost Calculation

2. Structural Cost

Similar to Engine Cost, Structural Mass and Structural Cost of a rocket also follow a proportional relation

$$C_S = M_S * (C_{Sss} / M_{Sss})$$

C_{Sss} = 75 million $

M_{Sss} = 760,000 kg

Cost Calculation

3. Propellant Cost

The propellant that we'll be using in our theoretical model will be liq. H_2 (1/7) and liq. O_2 (6/7)

$$C_P = (M_P * C_{O2} * (6/7)) + (M_P * C_{H2} * (1/7))$$

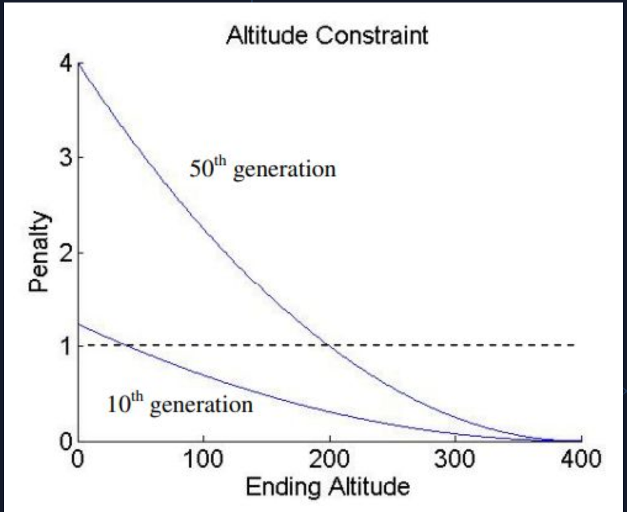
$C_{O2} = 0.155$ $/kg

$C_{H2} = 9.5$ $/kg

Penalty

A variable penalty is awarded to the rocket based on the maximum altitude it reaches i.e. A_{final}

$$p(A_{final}) = [abs(400 - A_{final})/ \max\{ 1, 400 - 4*gen \}]^2$$



Fitness

The fitness of the rocket is expressed as a function of 2 factors:

- Penalty (max Altitude)
- Number of dominance (n_{dom})

$$F = \max(1.0 - 0.01 * n_{dom} - p(A_{final}), 0)^2$$

Here, the more fit a model is the better, so we sort the models in ascending order based on [-1*fitness]

Generation 1

As seen in the last session, 5 parameters are used to define a rocket model.

But variation are required for comparison. Hence, we declare a “range” of possible values instead of a fixed number.

```
THRUST_VECTOR_RANGE = [
    [10.1e6, 10.3e6],
    [ 9.7e6,  9.9e6],
    [10.0e6, 10.2e6],
    [ 3.7e6,  3.9e6],
    [   2e3,    4e3]
]
ANGLE_VECTOR_RANGE = [
    [  350,   450],
    [320e3, 320e3]
]
CONE_HALF_ANGLE    = [ 0.10,  0.15]
ROCKET_RADIUS      = [  3.8,   4.2]
TOTAL_MASS         = [950e3, 980e3]
```

Next Generation

The next generation “inherits” the “genes” of the best of the previous population. Here, the genes correspond to the 5 different variables used to define the rocket model.

There is also a possibility of mutation as well, resulting in a change for better (as the worse one will be discarded anyway)

Python

In order to make it easier to maintain code, let’s define a few classes:

- RandomVariable
- Rocket
- Generation

NOTE: The library *pygmo* isn’t supported on windows, so you can use Google Collab or any other platform you feel comfortable using. It can be installed via:

```
!pip install pygmo
```

Simulation Results (DSTO)

Rocket Type	Actual Payload Mass (kg)[5]	Calc. Payload Mass (kg)	% Error	Actual Cost ($M)[5]	Calc. Cost ($M)	% Error	Actual m_p/C (kg/$M)[5]	Calc. m_p/C (kg/$M)	% Error
Delta II 7326-10	2294	2700	17.70	43.5	41.3	-5.06	52.74	65.38	23.97
Delta II 7925-10	4104	3500	-14.72	55	52.5	-4.55	74.62	66.67	-10.66
Delta IV Medium	8501	9990	17.52	72.6	70.1	-3.44	117.09	142.51	21.71
Delta IV Medium+ (4,2)	11455	13800	20.47	98	93.8	-4.29	116.89	147.12	25.87
Delta IV Heavy	21892	24500	11.91	150	162.3	8.20	145.95	150.96	3.43