Multi-Objective Low Thrust De-Orbit with Genetic Algorithm

Liam C. Smith
Ann and H.J. Smead Aerospace Engineering Sciences
University of Colorado Boulder
liam.smith@colorado.edu

Abstract*—Low Earth Orbit (LEO) is quickly becoming congested with large constellations of satellites, raising concerns over space safety and sustainability. The Federal Communications Commission (FCC) has released a fact sheet on "Mitigation of Orbital Debris in the New Space Age" which encourages spacecraft operators to both actively dispose satellites in LEO through atmospheric reentry and maintain active risk mitigation capabilities (maneuverability) through all phases of the mission [1]. Deorbit itself poses a multi-objective problem with potentially conflicting objectives; minimize the time it takes to deorbit (space safety) and minimize the amount of fuel (cost) to the operator. For spacecraft equiped with low thrust capabilities, this is often solved with an aero-assisted deorbit strategy which consists of gradually lowering perigee while allowing atmospheric drag to assist in lowering apogee with no additional cost to the operator. However, the FCC recommendation to maintain active risk mitigation capabilities throughout deorbit can be considered a constraint on this deorbit strategy. Lowering perigee too far into the atmosphere will cause the spacecraft to lose attitude control, losing the ability to perform any more deorbit maneuvers and violating the FCC recommendation. Thus a new multi-objective problem is born; minimize the time and fuel costs for a low thrust from deorbit subject to a minimum altitude for thrusting. This problem is analyzed and solved with a genetic algorithm.

***Index Terms*—Deorbit, Aero-Assist, Genetic Algorithm**

I. Introduction

With the increased utilization of space systems in Low Earth Orbit (LEO) since the turn of the century, deorbit has become highly studied mission phase. The community is evolving away from the old "25-year rule," a loose guideline encouraging LEO missions to design for a passive atmospheric reentry within 25 years of the end of mission, and towards more active and intentional deorbit techniques that account for space safety and sustainability [1]–[3]. One of these techniques is called an aero-assisted deorbit strategy, a strategy that utilizes atmospheric drag to aide in deorbit. But there are many ways to design an aero-assisted deorbit strategy and it can be complex to determine the optimal strategy. One technique to analyze and solve for the optimal aero-assit deorbit strategy is a genetic algorithm [4], [5]. A genetic algorithm is a global optimization technique that is inspired by the evolutionary concept of natural selection where only the fittest samples survive to create candidate solutions for the next iteration. While genetic algorithms do not guarantee an optimal solution, they continually improve generation to generation, providing insight into the objective space that drives the optimization problem.

II. Background

A. Atmospheric Drag

Atmospheric drag is a significant non-conservative force in LEO, which removes energy from an orbiting spacecraft causing the orbit to decay. While drag is typically an burden that must be managed during a LEO mission to remain in orbit, it becomes a useful asset for deorbit at the end of the mission. The acceleration due to drag is as follows,

$$\vec{a}_{drag} = -\frac{1}{2}C_d\frac{A}{m}\rho v_{rel}^2\ \hat{v}_{rel}$$

C_d : Coefficient of drag
$\frac{A}{m}$: Area-to-mass ratio (m^2/kg)
ρ : Atmospheric density (kg/m^3)
v_{rel} : Relative wind speed (m/s)
$\hat{v}_{rel}$: Relative wind velocity unit vector (m/s^2) (1)

Unless the spacecraft is a sphere of uniform material, both C_d and A vary with time as the cross sectional area facing the relative wind direction changes. Additionally, the atmospheric density, ρ, is a function of the orbital altitude and the space environment. Finally, the mass, m, may also change if the spacecraft is using propellant to maneuver. In general, the most challenging aspect to model is the atmospheric density due to the complex interactions between temperature, pressure, magnetic forces, and more which lead to considerable uncertainties. For this analysis the Harris-Priester density model is used because it is both smooth and computationally inexpensive, making it ideal for use in optimization applications [6].

B. Low Thrust De-Orbit Maneuvers

One factor that has made space much more accessible to all is the vast increase in space-grade, commercially available spacecraft components. With this has come an explosion in research and development in low thrust propulsion systems. Living up to their name, low-thrust systems deliver thrust on the order of tens to hundreds of milinewtons. This is almost negligible compared to their high energy counterparts; traditional liquid and solid rocket engines deliver thrusts upwards of thousands of newtons or more. The low thrust systems compensate for the lack of force by being highly efficient, delivering specific impulse values of well over 2,000 seconds. For comparison, liquid and solid rockets have specific impulse values on the order of 200 seconds [7]. In the modern space age, efficiency is prized over power and nearly all of the modern LEO satellites are equipped with low thrust propulsion systems.

Since low thrust maneuvers result in inherently small forces upon the spacecraft, any significant maneuver requires a long duration of thrust. The Gauss Variation of Parameters (VOP) method provides equations to analyze how a perturbing forces affect the orbital parameters. The one specific equation of interest for deorbit is the equation for the time rate of change of the semimajor axis, which is given in Vallado as [8]:

$$\frac{da}{dt} = \frac{2}{n\sqrt{1-e^2}}\left[e\sin{(\nu)}F_R + \frac{p}{r}F_S\right]$$

n: mean motion (s^{-1})
e : eccentricity
ν : true anomaly (rad)
a : semimajor axis (m)
F_R : acceleration, R component RSW frame (m/s^2)
F_S : acceleration, S component RSW frame (m/s^2)
(2)

Recognizing that the eccentricity and sine term attached to the F_R acceleration component are very small for nearly circular orbits, it becomes clear that the most efficient way to decrease the semimajor axis of an orbit (the requirement for deorbit) is to apply the low thrust acceleration in the negative F_S direction, meaning the anti-velocity direction. This matches intuition because a satellite must remove energy to deorbit. The final important observation for low thrust deorbit maneuvers comes from the Gauss VOP time rate of change equation for eccentricity [8]:

$$\frac{de}{dt} = \frac{\sqrt{1-e^2}}{na}\left[\sin{(\nu)}F_R + \left(\cos{(\nu)} + \frac{e+\cos{(\nu)}}{1+e\cos{(\nu)}}\right)F_S\right]$$

n: mean motion (s^{-1})
e : eccentricity
ν : true anomaly (rad)
p : semi-parameter (m)
r : orbital radius (m)
F_R : acceleration, R component RSW frame (m/s^2)
F_S : acceleration, S component RSW frame (m/s^2)
(3)

The key observation in Equation 3 is that if integrated over a single orbit period, $\frac{de}{dt} = 0$. This also follows intuition because always thrusting would result in equally lowering apogee and perigee, keeping eccentricity constant. On the other hand, thrusting for a partial orbit period results in a nonzero value for $\frac{de}{dt}$. From this it can be determined that centering a deorbit maneuver around apogee will increase eccentricity (effectively lowering perigee) and centering a deorbit maneuver around perigee will decrease eccentricity (effectively lowering apogee). This is a key component of choosing deorbit strategies.

III. De-Orbit Strategies

For this analysis, deorbit strategies are defined with a few basic assumptions. The first is that all deorbit maneuvers are low thrust retrograde maneuvers as described in the previous sections. The spacecraft will perform a single deorbit maneuver every orbit until the spacecraft reenters and is destroyed. The threshold for reentry is $r_a = 350km$, where r_a is the altitude of apogee. This may seem fairly high for an atmospheric demise, but this is a reasonable altitude for a satellite to lose attitude control due to increased atmospheric forces and be considered uncontrollable. This is as good as dead to an operator and is thus a reasonable termination condition. The second assumption is that deorbit maneuvers will be centered at apogee and the duration of the burn will be parameterized based on each specific deorbit strategy. A second parameter is the switching altitude. At this altitude, the deorbit maneuver will switch to be centered around perigee and the duration of the burn will remain unchanged. The idea behind this strategy is that, as just mentioned, there exists some altitude where the spacecraft will lose attitude control. It is assumed that the spacecraft is uncontrollable as soon as perigee crosses this minimum maneuver altitude threshold. In the case where perigee is being lowered more quickly than apogee, the switching altitude will allow for continued maneuverability until atmospheric drag naturally lowers perigee below the threshold altitude. The threshold for maneuverability is $r_p = 350km$, where r_p is the altitude of perigee. By defining deorbit strategies in this manner, a strategy can be completely defined by two parameters, the burn duration and switching altitude. The orbit averaged throttle profile (assuming the throttle is binary on or off) is constant for every orbit. All other factors such as thrust magnitude, drag area, and others remain constant between deorbit strategies.

A. *Passive De-Orbit*

The most simple deorbit strategy is a passive orbit. This strategy is to effectively do nothing, and allow the natural parasitic affect of atmospheric drag to remove orbital energy from the spacecraft until the satellite reenters and is destroyed. This strategy can be characterized with a burn duration of zero, making the throttle profile always zero. The switching altitude is irrelevant in this strategy because thrust is always off. While this case is not particularly interesting, it is a useful bounding case for the upcoming optimization. The altitude and throttle profiles for the passive deorbit strategy are shown in Fig. 1. This strategy also happens to be fuel optimal, minimizing the amount of fuel being used to deorbit with no regard to the deorbit timeline.

B. *Circular De-Orbit*

Another relatively simple deorbit strategy is the circular deorbit strategy. This strategy is the opposite of the passive deorbit strategy in that the burn duration is equal to the orbit period. The throttle profile of this strategy is always one and the switching altitude is again irrelevant. These are the only two strategies where this will be the case. This strategy

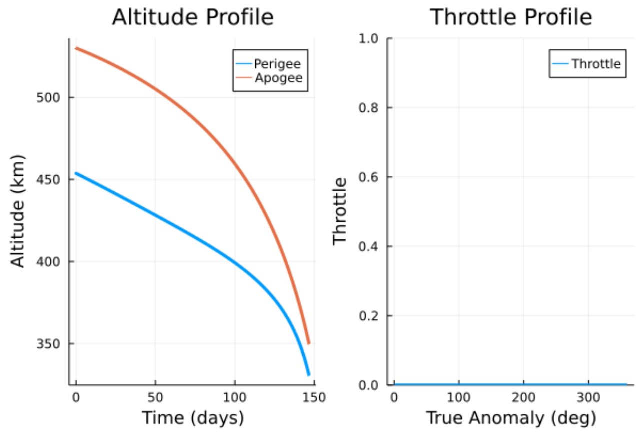

Fig. 1. The altitude and throttle profiles for the passive deorbit strategy.

is another useful bounding case because it is time optimal, minimizing the time it takes to deorbit with no regard to fuel usage. The altitude and throttle profiles for the passive deorbit strategy are shown in Fig. 2.

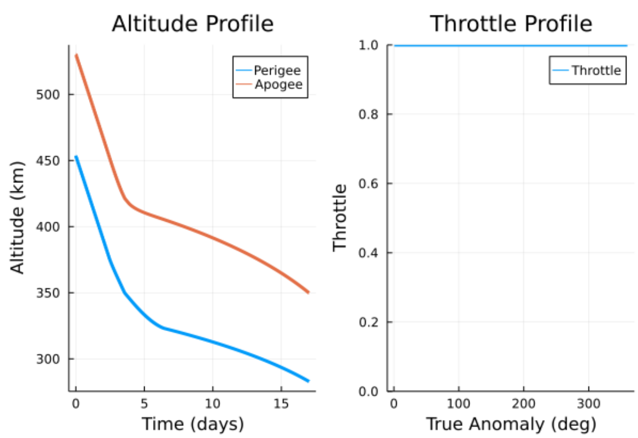

Fig. 2. The altitude and throttle profiles for the circular deorbit strategy.

C. Perigee Reduction De-Orbit

A more complex deorbit strategy is to save fuel by only lowering perigee. This strategy lowers perigee into the thicker parts of the atmosphere where atmopheric drag is significantly higher. This ”aero-assist” deorbit strategy can have any burn duration between 0 and 1 orbit period and the switching altitude becomes very relevant to the deorbit profile. The optimal values of both the burn duration and the switching altitude are the subject of this optimization problem. Using a switching altitude of 250km (this is notably less than the maneuver cutoff threshold altitude) and a burn duration of 1/16 of an orbit period (centered around apogee) results in the altitude and thrust profile in Fig. 3. Note that both the total fuel consumption and total time to deorbit will be greater than their respective optimal boundary cases.

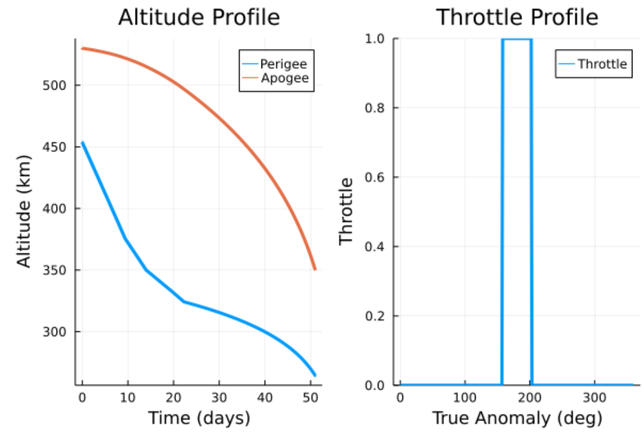

Fig. 3. The altitude and throttle profiles for the aero-assist deorbit strategy with burn duration = $pi/8$ radians and switching altitude = 250km.

IV. Problem Formulation

A genetic algorithm is an optimization technique that is inspired by the evolutionary concept of natural selection where only the fittest samples survive to pass on information to candidate solutions of the next generation [9]. These algorithms encode parameters of a problem into chromosome-like data structures and perform recombinations of selective pairs of ”parents” to create genetically similar ”offspring” which retain the most critical genes as measured against a fitness function. The methods of recombination, parent pair selection, offspring generation (often called crossover), and other minor hyperparameters are what define one genetic algorithm from another. The basic workflow of a genetic algorithm is to define an initial population, evaluate the fitness of the population, select the most fit candidate solutions for recombination, create the next generation of the population, and repeat. The genetic algorithm converges when either the average fitness or best fitness fails to improve after a number of generations.

A. The De-Orbit Genome

An individual candidate solution is defined by the deorbit genome, $X_{deorbit}$. The deorbit genome contains only the two parameters previously defined: the switching altitude and the burn duration, as seen in Equation 4.

$$X_{deorbit} = \begin{cases} \text{Switching Altitude } [250, 450]\ km \\ \text{Burn Duration } [0, \pi/2] \end{cases} \tag{4}$$

The switching altitude is a direct implementation of the parameter it represents with units of meters (sometimes presented in kilometers for easy reading). The switching altitude is bounded by the initial satellite altitude and the demise altitude. The burn duration is implemented as an angular width of Mean Anomaly in radians. This may seem slightly counter-intuitive at first, but it is very computationally frugal to calculate the Mean Anomaly from the satellite’s state vector and compare to the Mean Anomaly of the burn center (π for apogee or 0 for perigee). This results in a lower bound on the burn duration of 0. The upper bound on burn duration is set to $\pi/2$. This is specifically not set to π, which would be a forced circular deorbit, because that could introduce a potential decoupling between the two genes in this genome. This is to say, the satellite can achieve a circular deorbit by maximizing the switching altitude and the burn duration. This is because continuously lowering apogee reduces to a circular deorbit as apogee is constantly redefined when lowered below perigee. Using $\pi/2$ as the upper bound instead of π removes an entire quadrant of repeated solutions, simplifying the search space significantly. In this implementation, the values of each "gene" are the continuous parameters that they represent. This is in contrast to the canonical genetic algorithm, which calls for a binary representation of each gene [9]. This choice of a continuous representation of each gene will become more clear when the crossover function is described.

B. Deorbit Fitness Functions

The fitness function is a simulation of the deorbit strategy until demise as defined by the genome. The initial state of the the simulated satellite (in both classical orbital elements and Earth Centered Inertial (ECI) cartesian coordinates) can be found in Equations 5 and 6.

$$COES_0 = \begin{cases} \text{Semimajor Axis} : 6878000\ m \\ \text{Eccentricity} : 0.005 \\ \text{Inclination} : 69\ deg \\ \Omega : 55\ deg \\ \omega : 180\ deg \\ \text{Mean Anomaly (MA)} : 100\ deg \end{cases} \tag{5}$$

$$RV_0 = \begin{cases} \text{R} : [2712241.37, -358426.60, -6318958.88]\ m \\ \text{V} : [3891.70, 6403.93, 1265.18]\ m/s \end{cases} \tag{6}$$

The satellite is propagated using the SatelliteDynamics.jl Julia package with minor modifications to allow for thrust and stopping criteria [10]. The package uses a simple Cowell's Method of orbit propagation under the user provided throttle function. The only forces used in this analysis are atmospheric drag and thrust. The satellite parameters are described in Equation 7.

$$\text{Satellite} = \begin{cases} \text{Mass} : 500\ kg \\ \text{Drag Area} : 25\ m^2 \\ \text{Thrust} : f_{throttle}(MA) * 100\ mN \\ \text{Demise} : r_a < 350\ km \\ \text{Thrust Cutoff} : r_p < 350\ km \end{cases} \tag{7}$$

Once the satellite has crossed the demise threshold, the simulation ends and the candidate solution is evaluated for fitness. The fitness function is a linear combination of two objectives, the total deorbit time and the total time spent thrusting. This can be found in Equation 8.

$$\text{Fitness} = W_t * (t_f - t_0) + W_{th} * \sum_{i=0}^{N_f} f_{th}(MA_i) \tag{8}$$

In Equation 8, $f_{th}(MA_i)$, is the throttle function, which returns either 1 or 0 depending on the burn duration gene from the candidate solution. The fitness function also has two weight parameters, one for the time objective and one for the thrust objective. These weights serve two purposes. The first is to normalize the objectives to the same scale. The time objective is normalized by the passive deorbit time and the thrust objective is normalized by the sum term as evaluated in the circular deorbit strategy. The second function of the weights is to prioritize one objective over the other in a multi-objective fashion. This will be explored further in the next sections.

C. Selection, Crossover, and Mutation

The selection criteria for this genetic algorithm is an inverse roulette function. This means that each member of the current population is weighted by the inverse of the fitness value, and then are selected at random with replacement for recombination. This means that individuals with lowest combination of deorbit time and total thrust are considered the most fit.

The recombination function is simply the weighted average of the two parents. Since there is replacement in the selection process, the most fit individuals can recombine with itself to move on to the next generation without modification. In addition, a few of the most elite candidates (in this case only two) bypass recombination all together and simply move on to the next generation. The total population size is maintained through the recombination process.

The mutation function is simply a random selection of a new value from a uniform distribution between the bounds of the gene. Each individual has a 1% chance of mutation in each generation, and when mutating, each gene has a 50% chance of being mutated. This adds a nontrivial amount of exploration to the optimization space without dramatically slowing down convergence. Because the recombination function is an average of the two parents, exploration happens naturally, so mutation rates are kept fairly low.

Finally, the hyperparameters are chosen to balance gene diversity with computational load. The population size is set to 50, with convergence criteria of three generations without significant improvement. The initial population is a random selection of individuals from a uniform distribution within the bounds of each gene.

V. Results

The genetic algorithm is run 13 times, varying the relative weight between the two objectives between 0% and 200% to represent scenarios where the thrust objective is complete ignored (0%) and where the thrust objective is doubly important (200%). Before analyzing all the results, these edge cases can be used as sanity checks against the known solutions for each single objective.

Setting the thrust objective weight to 0 results in Figure 4. In this figure the fitness is reevaluated with a weight of 1 in order to compare with other solutions. Immediately, the genetic algorithm finds a solution that improves upon the naive circular deorbit strategy for the case where the amount of thrust used is completely irrelevant. The algorithm converges the solution: [422.581 km , 1.357 rad].

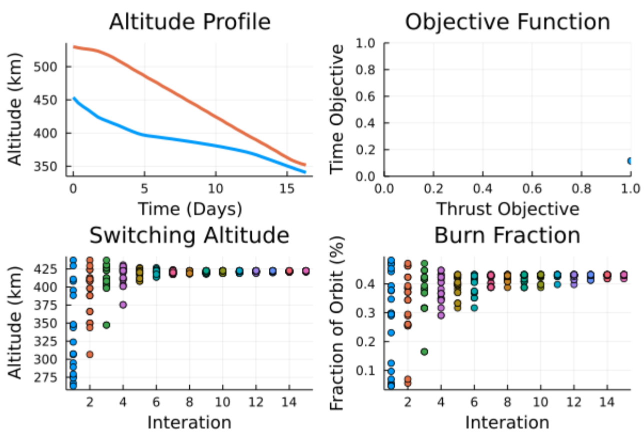

Fig. 4. Generic algorithm results for $W_t = 1.0$ and $W_{th} = 0.0$. The final fitness value is 1.11 with a genome of [422.581 km , 1.357 rad]. The altitude profile vs time (Top Left), the reevaluated (weights equalized) fitness value of the optimal solution (Top Right), the iteration history of the switching function gene (Bottom Left), and the iteration history of the burn duration gene (Bottom Right).

Analyzing this solution and referencing the altitude profile, it becomes clear that this strategy revolves around keeping perigee as high as possible (near maximum switching altitude) while focusing on lower apogee. This results in a circular deorbit, but one that is able to thrust for a bit longer than the naive circular deorbit strategy, which loses the ability to thrust as soon as perigee reaches the cutoff altitude. This solution may be considered a safer solution as per the FCC fact sheet as the deorbit time is shorter and the satellite maintains maneuver capability for a longer duration than the naive circular deorbit.

Setting the thrust weight to 2.0 (200%) represents an overvaluing on thrust relative to the deorbit time and is expected to result in the passive deorbit strategy. The results from the genetic algorithm are shown in Figure 5 and are almost identical to the passive deorbit strategy from the previous sections. The optimal genome for this solution is [335.109 km , 0.001 rad] with a final fitness value of 1.01. This result converged to essentially never turning the thruster on, which makes perfect sense given the extreme weight on the the thrust objective.

With these two bounding cases validating the genetic algorithm method, 11 more cases are run varying the thrust objective weight. The optimal solutions are found and all

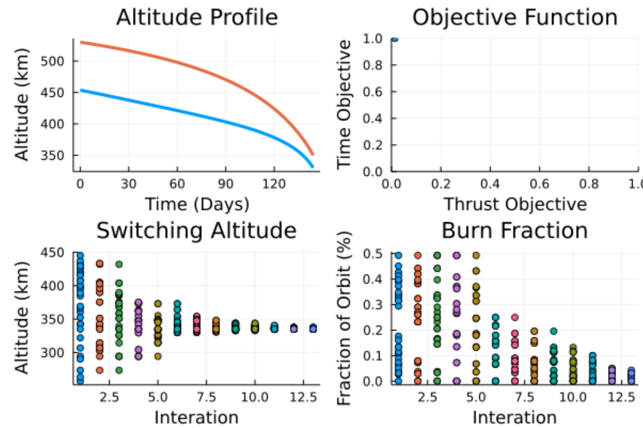

Fig. 5. Generic algorithm results for $W_t = 1.0$ and $W_{th} = 2.0$. The final fitness value is 1.01 with a genome of [335.109 km , 0.001 rad]. The altitude profile vs time (Top Left), the reevaluated (weights equalized) fitness value of the optimal solution (Top Right), the iteration history of the switching function gene (Bottom Left), and the iteration history of the burn duration gene (Bottom Right).

of the fitness values are reevaluated with neutral weights for comparison. The resulting figure, Figure 6, is the pareto front that describes the objective space of this deorbit problem.

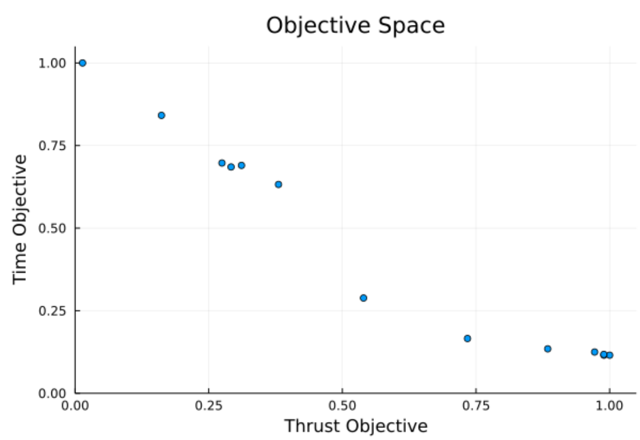

Fig. 6. The pareto front of the deorbit problem. Each point is an optimal solution representing a different relative weight of the time objective (y-axis) to the thrust objective (x-axis).

We quickly see our endpoints right where we expect them, and we see a linear trend emerge from the results. This is not surprising considering that the total objective is a linear combination of the two weighted objectives. However, with a closer look, there are some interesting points to consider. First, there seem to be two linear fits emerging from the sets of data. This would imply that the two objectives vary at different rates with respect to the independent variables (the genes), and one clearly dominates the other. These two linear fits can be seen in Figure 7. These linear fits are representative of α-vectors as seen in policy solutions to partially observable markov decision problems (POMDPS) and perhaps share a similar meaning [11]. Clearly, not all of the points on the green fit are physically possible, so there must exist some boundary constraints that are yet to be discovered in this objective space.

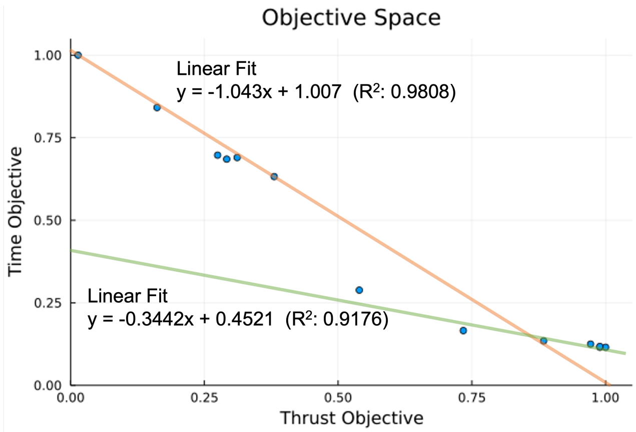

Fig. 7. The pareto front of the deorbit problem with two linear fits fit to subsets of the data. The green line (thrust focused fit) dominates the orange line (time focused fit) for most of the objective space. The intersection point is approximately [0.79, 0.18]

The overall fittest solution across all scenarios is shown in Figure 8. This solution is generated with the genetic algorithm with equal weight on the time objective and the thrust objective. The resulting trajectory is undoubtedly a perigee lowering deorbit strategy. The burn duration converges to about one-third of an orbit period, which, when centered around apogee like it is, is very efficient at lowering perigee. The switching altitude converges to below the thrust cutoff altitude, meaning there is no apogee lowering in this strategy. This suggests that the aero-assist perigee lowering deorbit is the most optimal combination of operator maneuvering and passive exploitation of atmospheric drag.

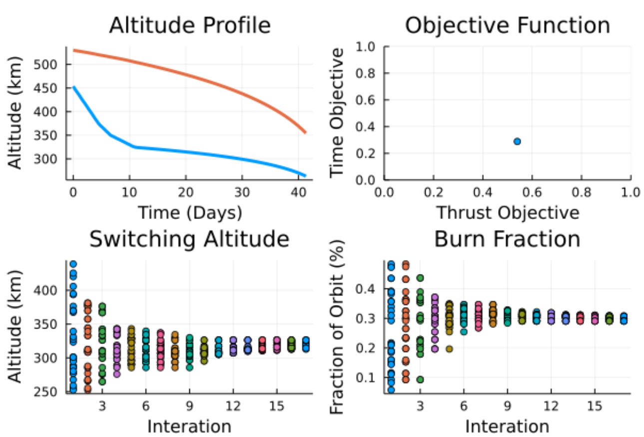

Fig. 8. Generic algorithm results for $W_t = 1.0$ and $W_{th} = 1.0$. The final fitness value is 0.83 with a genome of [326.276 km , 0.913 rad], which is the most fit across all tested combinations of weights. The altitude profile vs time (Top Left), the reevaluated (weights equalized) fitness value of the optimal solution (Top Right), the iteration history of the switching function gene (Bottom Left), and the iteration history of the burn duration gene (Bottom Right).

VI. Conclusion

There is a lot more to be studied in the area of space safety and sustainability, but it appears that the optimal strategy for deorbit from LEO is well understood. It is well known that an aero-assisted deorbit is beneficial in terms of saving fuel, but it is now confirmed that it remains very near optimal even when total deorbit time is considered. The genetic algorithm performed a broad search over the multi-object space and has now shown that when adopting a perigee lowering strategy, there is little to gain by stopping perigee lowering early and lowering apogee. The logical conclusion is to follow the perigee lowering strategy and only stop to lower apogee if there is excess fuel at the end of the mission. In future work, it would be ideal to confirm these conclusions with various starting orbit conditions. LEO extends up much further than the 500km starting orbit in this analysis, so perhaps there is more of the objective space to explore as the initial orbit gets higher and higher.

Appendix

All project code and example Jupyter notebooks can be found at https://github.com/lsmith7661/GeneticDeorbit.

References

[1] Federal Communications Commission, “Mitigation of orbital debris in the new space age,” *IB Docket No. 18-313*, April 2020.

[2] W. H. Ailor and R. P. Patera, “Spacecraft re-entry strategies: Meeting debris mitigation and ground safety requirements,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 221, no. 6, pp. 947–953, 2007.

[3] J. L. Rhatigan and W. Lan, “Drag-enhancing deorbit devices for spacecraft self-disposal: A review of progress and opportunities,” *Journal of Space Safety Engineering*, vol. 7, no. 3, pp. 340–344, 2020. Space Debris: The State of Art.

[4] WildArt, “Evolutionary.jl: Evolutionary genetic algorithms for julia v0.11.1.” https://github.com/wildart/Evolutionary.jl, 2022.

[5] Z. Michalewicz, C. Z. Janikow, and J. B. Krawczyk, “A modified genetic algorithm for optimal control problems,” *Computers Mathematics with Applications*, vol. 23, no. 12, pp. 83–94, 1992.

[6] N. Hatten and R. P. Russell, “A smooth and robust harris-priester atmospheric density model for low earth orbit applications,” *Advances in Space Research*, vol. 59, no. 2, pp. 571–586, 2017.

[7] A. Tummala and A. Dutta, “An overview of cube-satellite propulsion technologies and trends,” *Aerospace*, vol. 4, 12 2017.

[8] D. A. Vallado, *Fundamentals of Astrodynamics and Applications 4th ed.* Hawthorne, CA: Microcosm Press, 2013.

[9] D. Whitley, “A genetic algorithm tutorial,” *Department of Computer Science, Colorado State University*, 11 1993.

[10] D. Eddy, “SatelliteDynamics.jl,” 2021.

[11] R. D. Smallwood and E. J. Sondik, “The optimal control of partially observable markov processes over a finite horizon,” *Operations Research*, vol. 21, pp. 1071–1088, 09 1973.