Comparing Different Off-the-Shelf Optimizers' Performance in Conceptual Aircraft Design

Andrew D. Wendorff,* Emilio Botero,† and Juan J. Alonso‡

Stanford University, Stanford, CA 94305, USA

This paper covers the comparison of multiple optimization algorithms applied to SUAVE, a conceptual aerospace vehicle design tool. Using an aircraft similar to an Embraer E-190 as the initial conditions, we compare the optimum aircraft created and the convergence criteria of different optimizers. Our comparison is not only to find what optimizer works best for a specific problem, but to better understand the convexity, multi-modality, and discontinuities of the aerospace vehicle design space. We look at four different two variable optimization exploring aircraft geometry-geometry, geometry-mission, and mission-mission variable optimizations. In our studies, the surfaces did not have many local minimas so gradient-based optimizer found the same optimum point in approximately an order of magnitude less function evaluations than a population-based method. The full aircraft was also optimized. A little less than thirty percent reduction in design mission fuel burn was found by a gradient-based optimizer when changing only the geometry and an additional one percent was found by exposing the cruise segment mission parameters.

I. Introduction

Over the past few decades, numerical optimization has had an ever increasing impact on the design of commercial aircraft. How these optimization algorithms perform (how quickly they converge, what they perceive as optimum, how they deal with constraints, and the robustness of the potential convergence) can significantly impact the design process. Understanding the limitations of certain algorithms is critical to formulating reasonable problems. Both tube-and-wing and exotic configurations can be solved using the same mathematical formulation.

We are interested in understanding how SUAVE,[1] a conceptual design tool developed to analyze a variety of aircraft configurations, both traditional and exotic, performs utilizing a set of different optimizers. While SUAVE has the ability to examine a vast range of designs, in this study we will be focusing on a regional transport similar to the Embraer E-190.[2] SUAVE has been connected to multiple different optimization packages.[3] We are interested in understanding how optimal results and convergence performance vary due to different algorithms and package implementation of those algorithms. We will look at gradient based versus non-gradient based methods to identify the convexity, multi-modality, and discontinuities of the underlying function. Finally, we seek to understand the performance trade-off, as number of function evaluations, for incorporating mission variables in addition to vehicle variables.

The paper is constructed as follows. Section II contains a background of SUAVE and current analysis capabilities. Section III describes our initial conditions for the design along with constraints to make sure the vehicle generated is realistic and appropriately sized to meet commercial aircraft requirements. Section IV contains the optimization formulations for the different algorithms and will explain how the baseline problem has to be modified for different optimizers. Section V contains four two-dimensional problems to be optimized with all the different optimizers considered to visualize the design space. Section VI compares the performance of a single gradient based optimizer versus a population based method and how the optimal aircraft changes when mission variables are included. Finally, Section VII contains a summary of our results and conclusions about what optimizers work best with commercial transport aircraft in SUAVE.

*Graduate Student, Department of Aeronautics and Astronautics, AIAA Student Member.

†Graduate Student, Department of Aeronautics and Astronautics, AIAA Student Member.

‡Professor, Department of Aeronautics and Astronautics, AIAA Associate Fellow.

II. SUAVE Overview

SUAVE is currently a fully functioning aircraft conceptual design tool written in Python. Every major analysis discipline of aircraft design is represented in the code. A full break down of the analysis capabilities of SUAVE is described in a prior publication.[1] A partial list of the current capabilities is below:

- Aerodynamics for subsonic and supersonic flight,
- Weight correlations for tube-and-wing, blended wing body, and human powered aircraft,
- Segment based mission architecture,
- Static and dynamic stability metrics,
- Propulsion models for gas turbines and electric ducted fans,
- Energy networks for battery, fuel-cell, and solar based vehicles,
- Noise correlations for tube-and-wing aircraft,
- Basic performance estimation methods such as takeoff and landing field lengths and payload range diagrams,
- Interfaces to optimization packages, that require no changes to the optimization problem setup in SUAVE.

The open source nature of SUAVE lends the ability to contribute new analysis capabilities for future aircraft designs. Additionally, SUAVE was designed to produce outputs even when physically infeasible inputs are provided from an optimizer, something we will check. In our work, we will primarily incorporate less computationally expensive aircraft analysis tools since we are more interested in the optimization algorithms. All work presented here uses version 0.3.0 of SUAVE.[4]

III. Baseline Transport Aircraft

To compare different optimization algorithms, we need a reference aircraft from which to specify our initial conditions. We decided to use a standard tube-and-wing configuration as these methods inside SUAVE are most mature and most thoroughly developed. We will base our initial aircraft off of the Embraer E-190 and incorporate constraints necessary to produce a reasonable vehicle while minimizing the fuel burn of the aircraft over a 1000 nautical mile design mission.

A. Initial Aircraft

The initial aircraft is approximately the Embraer E-190.[2] For our study, we will use the variables defined in Table 1 where the reference aircraft initially burns 4391 kg of fuel over the entire mission.

We chose this set of parameters to fully define the vehicle and give the optimizer flexibility in creating new configurations while not incorporating so many variables that the studies take a prohibitively long amount of time to converge. Looking at the bounds for the design parameters, we set a relatively large range of parameters in order to test optimizer robustness and see the modality of the larger design space.

B. Design Objective and Constraints

We decided to optimize over fuel-burn as it is a plausible measure of the general performance of the aircraft. A 1000 nautical mile mission is a standard case employed to compare different aircraft. To make sure that we are creating a reasonable vehicle that is meeting the same design requirements aircraft manufacturers use, we added a set of constraints shown in Table 2.

Additional missions were simulated to ensure the aircraft could meet the varied requirements imposed by airlines. The first additional mission is the max range mission, of 2300 nautical miles. The second additional mission requires the vehicle takeoff from a constrained short field and cruise 500 nautical miles. By ensuring the vehicle meets the requirements for 3 separate missions provides reasonable assurance that the vehicle has practical use for commercial activities.

Table 1: Design Variables with Initial Conditions and Bounds

		Reference airplane	Lower Bound	Upper Bound
Wing Aspect Ratio	[-]	8.4	5	20
Wing Area	[m^2]	92	70	200
Wing Tip Twist	[°]	-2	-12	12
Wing Sweep	[°]	23	0	35
Wing t/c	[-]	0.11	0.07	0.2
Wing Taper	[-]	0.28	0	1
Design Thrust	[N]	37278	10000	70000
Maximum Takeoff Weight	[kg]	51800	20000	100000
Flap Takeoff Angle	[°]	20	0	40
Flap Landing Angle	[°]	30	0	40
Short Field Takeoff Weight	[kg]	45000	20000	100000
Design Takeoff Weight	[kg]	50000	20000	100000
Cruise Altitude	[km]	11	6	14
Cruise Velocity	[knots]	450	344.1	516.2

Table 2: Constraints for Optimization Study

Constraints	Bound		
Wing Span	$\leq$	118	[ft]
Takeoff Noise	$\leq$	92.5	[dB]
Sideline Noise	$\leq$	95.0	[dB]
Landing Noise	$\leq$	99.0	[dB]
Maximum Zero Fuel Weight Consistency	$>$	0	[-]
Design Range Fuel Margin	$>$	0	[-]
Short Field Fuel Margin	$>$	0	[-]
Maximum Range Fuel Margin	$>$	0	[-]
Takeoff Field Length	$\leq$	2056	[m]
Short Takeoff Field Length	$\leq$	1165	[m]
Landing Field Length	$\leq$	1700	[m]
Second Segment Climb Gradient Max Range	$\geq$	2.4	[%]
Second Segment Climb Gradient Short Field	$\geq$	2.4	[%]
Maximum Throttle	$\leq$	1	[-]

The constraints we added are a combination of geometric (wing span), performance (noise and takeoff lengths), and vehicle capabilities (maximum throttle and fuel margins) among others. We believe this set of constraints is large enough to allow the optimizer to pick reasonable aircraft, but small enough that an optimal aircraft can be found in a reasonable amount of time.

IV. Optimization Formulations

Starting from this baseline aircraft, we will be solving the generalized non-linear programming problem defined in Equation 1

$$
\begin{aligned}
\underset{x}{minimize} \quad & f(\mathbf{x}) \\
\text{subject to} \quad & g_j(\mathbf{x}) = 0 \quad j \in \{1, ..., l\} \\
& h_k(\mathbf{x}) \leq 0 \quad k \in \{1, ..., m\} \\
& lb_i \leq x_i \leq ub_i, \ i \in \{1, ..., n\} \\
& \mathbf{x} \in \mathbb{R}^n
\end{aligned}
\tag{1}
$$

where $\mathbf{x}$ are the design variables (both aircraft and mission) f is the objective, the fuel burn in our case, g_j are the equality constraints, h_k are the inequality constraints, lb_i and ub_i are the lower and upper bounds for each design variable i, respectively, and n is the total number of design variables.

Having set up a generalized design problem, we are interested in understanding how different optimization methods change the resulting aircraft and the required computational expense to generate those results. In order to do this, we will look at both gradient based methods to determine which algorithms converge rapidly and non-gradient based methods to try to identify how convex and multi-modal the design space is actually.

We have previously connected SUAVE to PyOpt, a Python-based optimization package containing multiple algorithms for solving nonlinear constrained optimization problems.[5] Additionally, SUAVE is integrated with SciPy, another Python-based optimization package. The SciPy connectivity enables access to other optimization algorithms, but we will only be using SLSQP from SciPy in this study.

A. Gradient Based Methods

1. *SNOPT*

SNOPT is a nonlinear gradient based optimization algorithm suitable for large problems with nonlinear constraints.[6] It is based on a sparse Sequential Quadratic Programming (SQP) method. SNOPT uses limited-memory quasi-Newton approximations to the Hessian of the Lagrangian. SNOPT is very robust and can even handle discontinuities provided they are not close to local minima.[6] SNOPT is coded in Fortran but interfaced with SUAVE through PyOpt. This PyOpt interface allows for parallel finite-difference gradients of the SUAVE function.

2. *SLSQP*

An SLSQP algorithm, or sequential least squares programming, is interfaced through PyOpt again. This algorithm is based on Kraft's 1988 work.[7] This SLSQP algorithm uses the Han-Powell quasi-Newton method with a BFGS update of the B-matrix. Again the PyOpt interface allows for finite difference gradient calculations.

Additionally, another version of SLSQP is available to SUAVE through SciPy. These two versions of SLSQP will be compared to see if the resulting optimum is substantially different from the other.

B. Non-Gradient Based Methods

1. *COBYLA*

COBYLA, constrained optimization by linear approximation, is a derivative free optimizer that does not account for bounds explicitly.[8] This algorithm works by creating a linear approximation of the objective and constraints for n+1 points in the design space that a simplex can work over to find the optimum in the trust region. The trust region radius is then modified as the algorithm moves towards the optimum.

2. *ALPSO*

The Augmented Lagrangian Particle Swarm Optimizer[9] (ALPSO) provides a population based optimizer reference to compare to the gradient based method results against. This algorithm will find global optima using a particle swarm method. The constraints are enforced through augmented Lagrange multipliers. This method is accessed through PyOpt using Dynamic Process Management that effectively parallelized the computations. When using this algorithm, the sign of the constraint values must be flipped compared to other algorithms in PyOpt.

V. Optimization Exploration

Before performing the full vehicle optimization discussed in Section III, we are starting with a variety of two-variable problems. Optimizing the initial aircraft within small two-variable cases allows us to better understand the convexity and multi-modality of design space and how different optimizers perform through visualization. The two variable design spaces considered are:

- Wing area and maximum takeoff weight
- Wing thickness-to-chord ratio and wing sweep
- Wing thickness-to-chord ratio and design mission cruise velocity
- Design mission cruise velocity and design mission cruise altitude

We chose these cases as some are standard studies considered in aircraft design. Other cases will help to determine if different optimizers perform better with different variable types: mission variables versus aircraft variables. Finally, we have an expectation on what the optimization surface should look like generally, but these cases give us the opportunity to better understand if these surfaces are multi-modal, convex, or otherwise non-linear where it might be difficult to find the global optimum when utilizing gradient-based algorithms. Since our goal is not to guarantee to find the global aircraft design optimum, but just to better understand how the aircraft optimization process would work in practice, we will only start our optimization from the baseline vehicle described in Section III in each of these studies. For all these cases, we will minimize the design fuel burn with a single constraint on fuel margin over said mission.

A. Wing Area - Maximum Takeoff Weight Optimization

The first two variable case we are considering is wing area versus maximum takeoff weight. We expect the smaller the wing, the small the fuel burn. In addition, for a lighter aircraft, less fuel is needed for the mission. The results generated from the different optimizers are contained in Table 3 and the optimization iterations are shown in Figure 1 where the circle denotes the starting point and the different color triangles denote what each optimizer believes is converged optimum. The contour map shows the fuel burn over the domain and the lines denote the fuel consistency constraint throughout the design space.

Table 3: Results for Wing Area - Maximum Takeoff Weight Optimization

Optimizer	Wing Area (m^2)	MTOW (kg)	Fuel Burn(kg)	Function Calls	Major Iterations
SNOPT	111.67	46818.6	4103.3	23	6
COBYLA	143.6	60000.0	4895.6	51	∼
SLSQP (SciPy)	111.66	46818.6	4103.3	24	6
ALPSO	115.4	46846.4	4106.6	520	∼

Looking at these results, we see SNOPT moves from the starting point and spends a large number of function calls on the fuel margin constraint boundary. To better illustrate the major steps taken in the SNOPT algorithm, Figure 2 shows the step-by-step progression of the major iterations to finally get to the optimum. It takes the six major iterations for SNOPT to fully converge. Most of the iterations are spent on the design range fuel margin constraint line identifying where along this surface the minimum actually occurs. From Figure 2d, we see SNOPT has found the general area of the optimum, but additional steps are taken to satisfy the constraints. The full progression is shown in Figure 2g.

The results for SLSQP implemented in SciPy show a different progression, as expected since it is solving a sequential least squares problem, to get basically the same optimum. SLSQP takes one function call more than SNOPT, but takes a different progression through the domain to find the optimum. Both SLSQP and SNOPT spend most of their major iterations along the constraint boundary. COBYLA takes significantly more function calls to get a worse resulting optimum. This is unexpected since COBYLA solves with a simplex method that should be robust to the optimum. The ALPSO particle swarm algorithm results shown in Table 3 approach the optimum found from SLSQP and SNOPT. The particle swarm takes more function

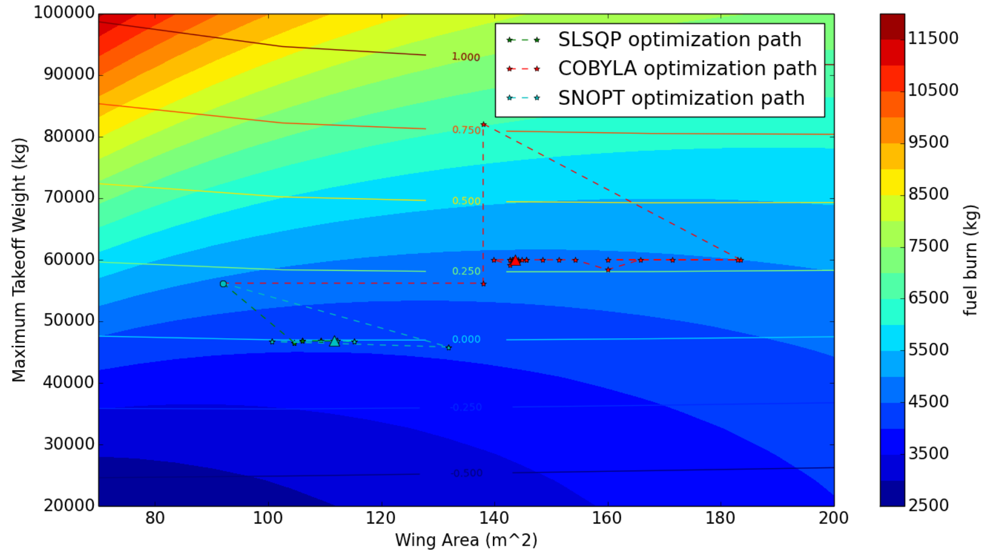


Figure 1: Wing Area - Maximum Takeoff Weight Contour Plot with Optimization Iterations

evaluations as expected since it does not evaluate gradients and in this two-dimensional domain, the gradient based optimizers do not get stuck in local minima.

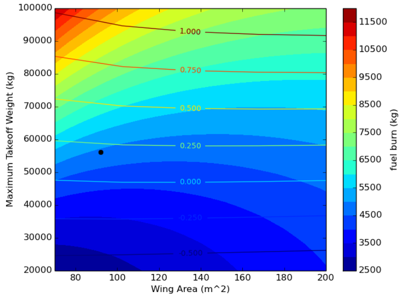
B. Wing Thickness-to-Chord Ratio - Wing Sweep Optimization

Beyond the results for the standard wing area - maximum takeoff weight aircraft optimization. Another interesting study in aircraft design is that of wing thickness-to-chord ratio versus wing sweep. This trade focuses on identifying the appropriate wing shape in cruise where our baseline aircraft is flying at Mach 0.785. We expect that as the wing thickness increases, the wing sweep should also increase to reduce the compressibility drag. Increasing the thickness of the wing will increase the moment of inertia thus decreasing the bending stresses experienced by the wing. This decrease in the stress seen by the wing means the structural weight can be reduced. Increasing the wing sweep will increase the weight of the wing due to an increase of a twisting moment, however. There is thus a trade between these two variables between the amount of fuel burned in flight due to compressibility drag and fuel burned due to a heavier aircraft.

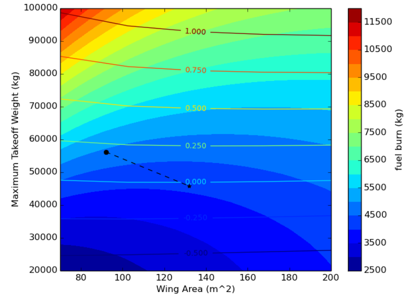
The results for this optimization are presented in Table 4. Both SNOPT and SLSQP find the optimum for the vehicle in one step as shown in Figure 3. Each optimizer finds the solution in one major iteration, but SNOPT solves the problem in one less function evaluation. We chose the upper and lower bounds for the design variables based on similar aircraft. Looking at the contour, however, we do not see the fuel burn increasing as we go to higher sweep values. This is a limitation of the current weight models incorporated within SUAVE. In the future, a higher fidelity model could change this characteristic. Since we are not trying to improve the capabilities in SUAVE, but instead identify how optimizers perform, this result satisfies our expectations.

Table 4: Results for Thickness-to-Chord Ratio - Wing Sweep Optimization

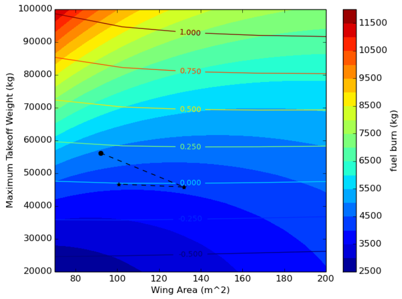
Optimizer	t/c (∼)	Wing Sweep (deg)	Fuel Burn(kg)	Function Calls	Major Iterations
SNOPT	0.07	40	3958.2	8	1
COBYLA	-0.246	20	3629.3	66	∼
SLSQP (PyOpt)	0.07	40	3958.2	9	1
ALPSO	0.07	40	3958.2	280	∼



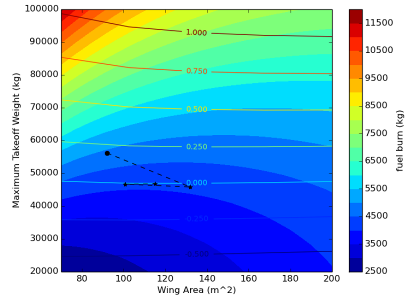
(a) Initial Starting Point



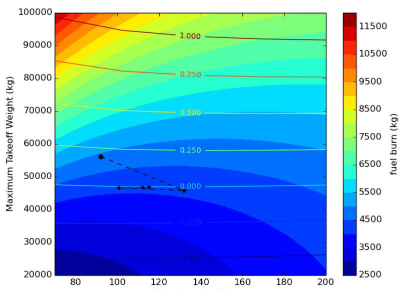
(b) First Major Iteration



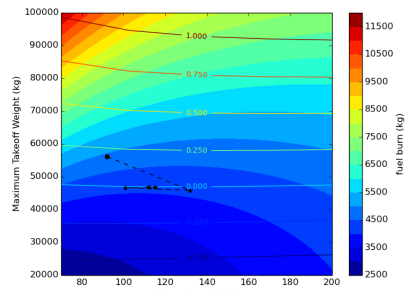
(c) Second Major Iteration



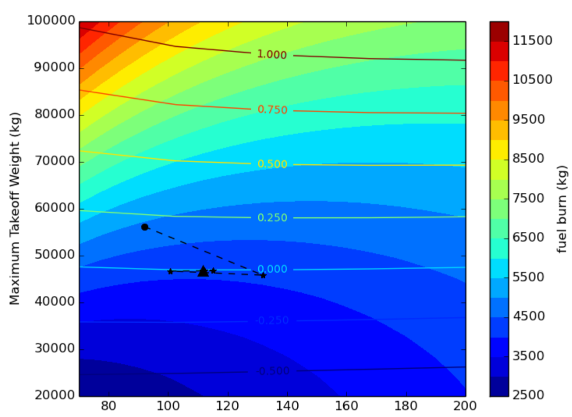
(d) Third Major Iteration



(e) Fourth Major Iteration



(f) Fifth Major Iteration



(g) Sixth Major Iteration

Figure 2: Illustration of Major Iterations of SNOPT Optimization in the Wing Area - Maximum Takeoff Weight Design Study

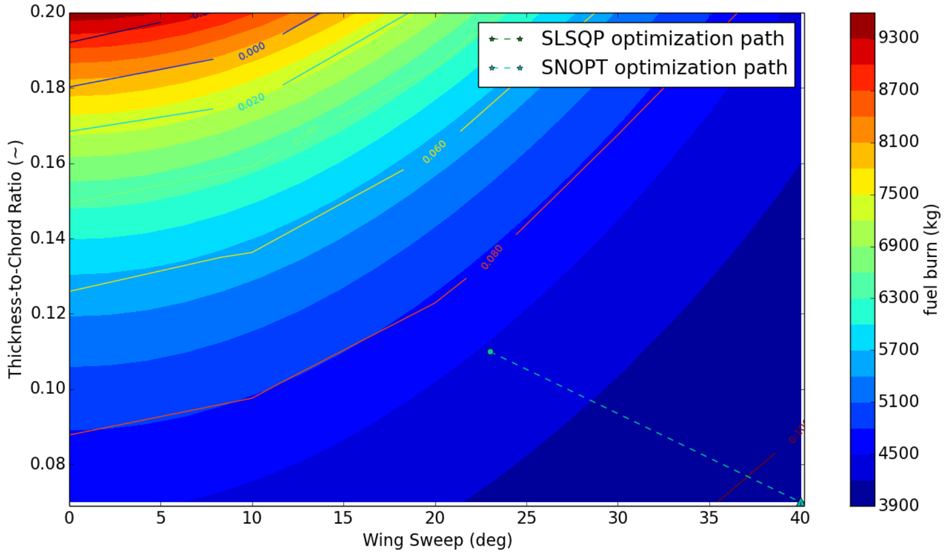


Figure 3: Wing Sweep - Wing Thickness-to-Chord Ratio Contour Plot with Optimization Iterations

What does not meet our requirements, however, are the results generated from COBYLA. COBYLA does not allow for bounds for design variables to directly specified in the formulation. To account for the bounds, we added four additional constraints. Obviously, the constraints were not all satisfied. While this is not ideal, it does illustrate how SUAVE has been developed to be robust to non-physical results and still produce values back to the optimizer. The ALPSO algorithm performs as it did in the wing area - maximum takeoff weight optimization. It takes a greater number of function evaluations to find a similar optimum as that found by SNOPT and SLSQP.

C. Wing Thickness-to-Chord Ratio - Design Mission Cruise Velocity

In addition to optimizing the aircraft based on just aircraft design parameters for a fixed mission, an aircraft design can be optimized for both aircraft parameters and mission variables simultaneously. These types of problems can dramatically expand the total number of design variables modified in creating the solution. The potential reward for an aircraft designed concurrently with a mission can be relatively large. We want to understand the effect a mission variable versus an aircraft variable has on the optimization. Are they completely interchangeable? Do mission variables require more function evaluations in general? Do they require less? The number of function evaluations will stem from the topography of the optimization surface. In Figure 3, the solution space was well behaved where SNOPT and SLSQP could find the optimum in one major iteration. Figure 4 is not as well behaved as illustrated by the optimization steps each optimizer takes. Again, COBYLA does not satisfy the constraints and creates an infeasible design. At present, we are not sure what is causing the constraints to not be satisfied in the COBYLA interface with PyOpt.

SNOPT and SLSQP, however, both find solutions right on the boundary of the domain. Each takes an optimization step to the corner of the domain, but unlike the wing sweep - wing thickness-to-chord ratio case, the optimum is not in the corner. From there, the two optimizers take different paths to find the solution with SNOPT performing better. SNOPT goes back to the center of the domain and then checks another corner. Finding the optimum is not in that corner either, then it starts exploring in the large area in the center where the optimum actually occurs. The results for this optimization are shown in Table 5. SNOPT takes 29 function evaluations while SLSQP took 35 evaluations. The ALPSO algorithm takes more function calls to get to the optimum, but finds the same solution just as the previous cases.

Where SNOPT went back to the center of the domain to take steps. SLSQP from PyOpt stayed on the boundary of the domain and actually evaluated non-physical points before recovering to approximately the same solution as SNOPT for a slight degrade in performance. If SUAVE had been less robust, then the solution would not have been found successfully. This case illustrates the necessity for conceptual aircraft

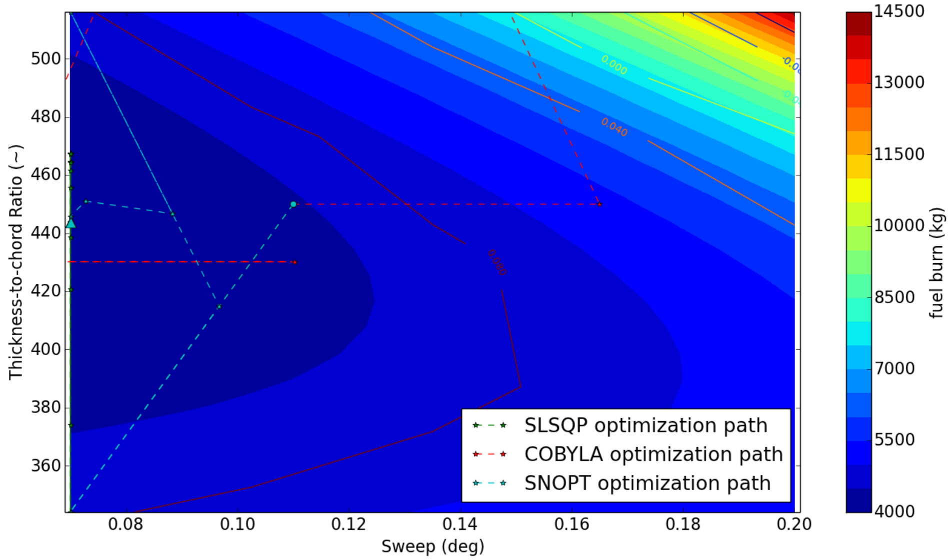


Figure 4: Wing Thickness-to-Chord Ratio - Design Mission Cruise Velocity Contour Plot with Optimization Iterations

Table 5: Results for Thickness-to-Chord Ratio - Design Mission Cruise Velocity Optimization

Optimizer	t/c (∼)	Cr. Velocity (knots)	Fuel Burn(kg)	Function Calls	Major Iterations
SNOPT	0.07	443.65	4167.63	29	8
COBYLA	0.105	0	3710.38	62	∼
SLSQP (PyOpt)	0.07	444.09	4167.65	35	4
ALPSO	0.07	443.56	4167.63	280	∼

tools to be robust to work with certain optimizers.

We have utilized different optimization packages with SLSQP for the results to this point. Table 6 shows the performance of the SLSQP algorithm in PyOpt versus SciPy for this case. The SciPy version of the algorithm solves in fewer function evaluations and finds a better optimum than the PyOpt version of the code. This result shows how the implementation of the same algorithm in different packages can create different vehicles in the end even on these simplified test cases. When comparing SUAVE results in the future, it is important to understand what is causing these discrepancies. Until then, not only the optimization algorithm, but also the package implementation must be specified for a consistent comparison.

Table 6: Comparing the SLSQP Algorithm Implemented in PyOpt versus SciPy

Optimizer	t/c (∼)	Cr. Velocity (knots)	Fuel Burn(kg)	Function Calls	Major Iterations
SLSQP (PyOpt)	0.07	444.09	4167.65	35	4
SLSQP (SciPy)	0.07	443.70	4167.63	25	6

D. Design Mission Cruise Velocity - Design Mission Cruise Altitude

The last two variable case we consider is a two mission variable set where we change the cruise velocity and the cruise altitude to find minimum fuel burn. As the aircraft flies higher, the atmospheric density and temperature both decrease causing a speed of sound decrease due to the temperature. If the aircraft flies higher, the drag can be decreased due to a lower density. There is a trade-off for the increased energy required to climb. After reaching a certain altitude, the benefit in climbing is lost. This results in a degradation in performance. In the same way as seen in the previous optimization for cruise velocity versus wing thickness-to-chord ratio, above a certain speed performance is decreased and the same is true below a certain speed. The contour with optimization iterations for all three of SLSQP, SNOPT, and COBYLA are shown in Figure 5. The constraint curves have been removed because the constraint is not active in this part of the design space. Also, notice that at a little over 11 km, there is a discontinuity in the fuel burn contour.

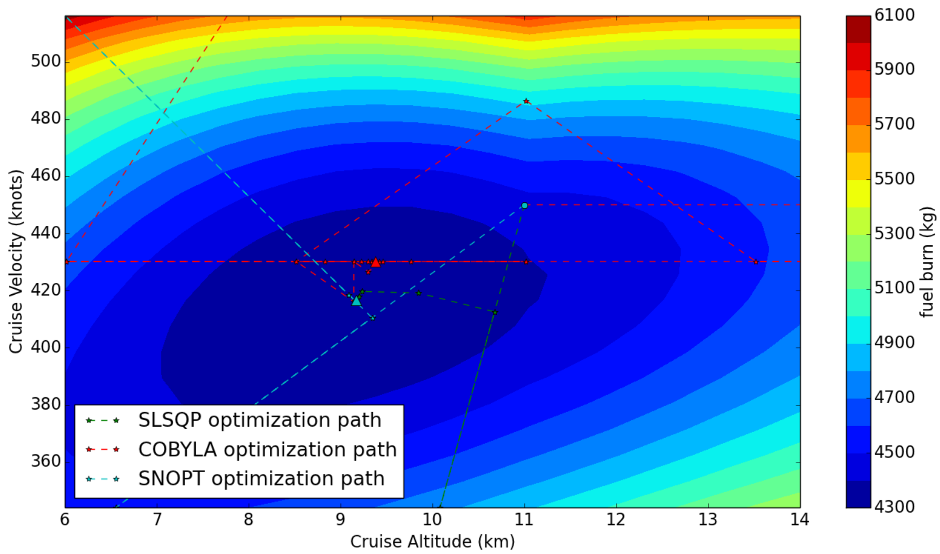


Figure 5: Cruise Altitude - Cruise Velocity Contour Plot with Optimization Iterations

In this case, we have a nice bullseye in our domain that SNOPT and SLSQP find. SLSQP actually finds it in less iterations than SNOPT as shown in Table 7. Both gradient based methods are not affected by the discontinuity in the objective function. COBYLA does not perform as well and still does not find the optimum even in this simplified case. It also requires significantly more function evaluations to solve to an inferior point. As expected, ALPSO also finds the optimum, but just takes more function evaluations.

Table 7: Results for Design Mission Cruise Velocity - Design Mission Cruise Altitude Optimization

Optimizer	Velocity (knots)	Altitude (km)	Fuel Burn(kg)	Function Calls	Major Iterations
SNOPT	416.62	9.17	4309.8	29	5
COBYLA	430.18	9.38	4322.85	48	~
SLSQP	416.62	9.17	4309.8	22	5
ALPSO	416.32	9.20	4309.8	280	~

Looking at Table 5 and Table 7 compared to Table 4, the mission variables seem to require more function evaluations to solve. This is not meant to be a generalized statement, but just an observation of what has been seen in these cases. The performance of SNOPT and SLSQP versus COBYLA is realized over the entire set of problems. It should be stressed that this is just COBYLA in PyOpt and not a generalized result for other optimization packages. Since COBYLA incorporates a simplex method, it should in general converge to the global optimum. The ALPSO algorithm performs well in finding the optimum over the set of test problems considered. It takes a longer time to find the solution, but is more likely to find the global optima. As such, we recommend utilizing SNOPT or SLSQP, preferably in SciPy, if these packages are available in future aircraft optimizations in SUAVE when the designer or engineer wants to find an improvement in the aircraft design quickly that is not necessarily the global optimum. Our two dimensional studies show that the aircraft design space seems to be relatively convex and thus well suited to gradient based methods. ALPSO can also be utilized if the aircraft function is cheap enough to run that the order of magnitude increase in function evaluations to find the optimum is still within the computational budget or it is critical to find the global optimum compared to the local optimum.

VI. Aircraft Results

After comparing the two dimension aircraft optimizations conducted in Section V, we want to optimize the baseline aircraft. We will start by keeping the mission fixed and only modifying the aircraft. After finding how the aircraft changes for a set mission, we will then evaluate the aircraft performance where just the design mission cruise conditions are changed. Finally, we will optimize the aircraft where all missions cruise conditions are variables.

A. Aircraft Variable Design Optimization

For the case where we are only varying the the aircraft characteristics, we will compare the performance of SNOPT versus ALPSO. Since SNOPT is a gradient based optimizer compared to the particle swarm ALPSO methodology, we expect that SNOPT will perform better unless the design problem is very multi-modal where ALPSO can escape to a better optimum region. The results for these two optimizations are shown in Table 8.

Looking at these results, we see that SNOPT has pushed up to the boundaries we expect designers to intuitively do also such as maximizing span and removing any excess weight from the vehicle. Just as we saw in the two dimensional cases, the sweep goes all the way to the maximum value allowed. The taper for the SNOPT result goes to the lowest value. This is another issue with the wing weight estimation that could be improved in the future. Interestingly, the ALPSO result actually does the opposite and pushes the wing taper to its maximum value. Population methods are useful in that they explore dramatically different regions resulting in some unexpected results. Some might make sense while others such as the short field takeoff weight being higher than the design takeoff weight even though the short field mission is shorter do not make sense, but the result is based on where the optimizer sampled results.

In this case, ALPSO had 2200 function evaluations and lowered the design fuel burn by approximately twenty-three percent from 4391 kg to 3385.2 kg. SNOPT performed better by decreasing fuel burn to 3196.7 kg utilizing only 751 function evaluations. As mentioned above, this result is expected for convex shapes. ALPSO was stopped by a function evaluation limit so we cannot guarantee that SNOPT found the global optimum, but this example illustrates just how powerful the gradient based optimizers are in finding relatively good optimizers in the aircraft design space. We see that the short field takeoff length is restricting the size of the engine for this methodology. Overall, these results are as expected and show the difference in results between gradient based and population based methods.

B. Aircraft and Mission Variable Design Optimization

When incorporating mission variables with the aircraft characteristics, we are going to limit mission variables to only the cruise segment. SUAVE is set up so other variables could be changed. We are interested in how the performance might be improved for expanding the design problem. The design mission fuel burn will be the metric of interest for the two cases considered. In one case, we will only vary the design mission cruise altitude and velocity leaving the max range and short field mission characteristics fixed at the values specified in Table 1. The second case expands the design space to vary all the different mission cruise segments separately to see if there is any improvement found in the design mission for varying flight conditions in other segments. Table 9 shows the results for both cases where there is a very slight improvement for the additional mission variables. Comparing these results with those in Table 8, we see that the improvement in fuel burn, approximately one percent, comes from exposing the design mission variables to the optimizer with an almost negligible change in optimum result for changing the design parameters of the other missions.

Table 8: Comparing Optimal Aircraft Generated from SNOPT and ALPSO

		SNOPT	ALPSO	Bounds
Wing Aspect Ratio	[-]	18.5	15.5	[5,20]
Wing Area	[m^2]	70.0	73.3	[70,200]
Wing Tip Twist	[°]	-0.511	-0.248	[-12, 12]
Wing Sweep	[°]	40.0	40.0	[0,40]
Wing t/c	[-]	0.112	0.136	[0.07,0.12]
Wing Taper	[-]	0.0	1.0	[0, 1]
Design Thrust	[N]	49270	49560	[20000, 70000]
Maximum Takeoff Weight	[kg]	49790	50390	[20000,100000]
Flap Takeoff Angle	[°]	21.9	39.96	[0, 40]
Flap Landing Angle	[°]	30.0	40.0	[0, 40]
Short Field Takeoff Weight	[kg]	44090	47560	[20000, 100000]
Design Takeoff Weight	[kg]	45620	45900	[20000, 100000]
Wing Span	[ft]	118.0	110.8	$\leq$ 118.0
Takeoff Noise	[dB]	92.48	92.49	$\leq$ 92.5
Sideline Noise	[dB]	89.97	89.99	$\leq$ 95.0
Landing Noise	[dB]	92.37	92.79	$\leq$ 99.0
Maximum Zero Fuel Weight Consistency	[-]	6.62e-14	1.28e-4	$\geq$ 0
Design Range Fuel Margin	[-]	-4.52e-13	3.15e-5	$\geq$ 0
Short Field Fuel Margin	[-]	-4.07e-13	0.0758	$\geq$ 0
Maximum Range Fuel Margin	[-]	-8.27e-12	1.67e-4	$\geq$ 0
Takeoff Field Length	[m]	1069.4	1222.0	$\leq$ 2056
Short Takeoff Field Length	[m]	1165	1106.4	$\leq$ 1165
Landing Field Length	[m]	1448.3	1554.6	$\leq$ 1700
Second Segment Climb Gradient Max Range	[-]	0.146	0.132	$\geq$ 0.024
Second Segment Climb Gradient Short Field	[-]	0.175	0.145	$\geq$ 0.024
Maximum Throttle	[-]	0.596	0.638	$\leq$ 1
Design Fuel Burn	[kg]	3196.7	3385.2	-
Function Evaluations	[-]	751	2200	-

The cost of finding these improvements is approximately 60 percent for the design cruise mission and 140 percent when exposing all the mission's cruise variables.

We chose to run all the results in SNOPT due to design space appearing to not being too multi-modal where a gradient-based methodology would perform poorly. The cause for this dramatic increase in SUAVE function evaluations to find the optimum is still being explored.

VII. Summary & Conclusions

Throughout this paper, we have shown how different off-the-shelf optimizers can be integrated with SUAVE in the conceptual design phase. SNOPT seems to be the best option, if available, to find an optimal aircraft. In the cases considered, the objective surfaces had a single optimum that could be found easily with gradient based methods. Depending on the type of aerospace vehicle to be optimized in the future, the surface might have multiple local minima that might make a non-gradient based method more appealing. When applying SNOPT to a full aircraft configuration, the fuel burn was decreased from 4391 kg to approximately 3200 kg if the aircraft geometry is varied and 3140 kg if both the geometry and mission are optimized. This

Table 9: Comparing SNOPT Optimized Aircraft for Design Mission Variables Versus All Mission Design Variables

		Design	All Missions	Bounds
Wing Aspect Ratio	[-]	17.44	17.42	[5,20]
Wing Area	[m^2]	74.15	74.26	[70,200]
Wing Tip Twist	[°]	-0.740	-0.761	[-12, 12]
Wing Sweep	[°]	40.0	40.0	[0,40]
Wing t/c	[-]	0.1076	0.1072	[0.07,0.12]
Wing Taper	[-]	0.0	0.0	[0, 1]
Design Thrust	[N]	49668.7	49683.8	[20000, 70000]
Maximum Takeoff Weight	[kg]	49799.4	49598.8	[20000,100000]
Flap Takeoff Angle	[°]	20.9	21.03	[0, 40]
Flap Landing Angle	[°]	30.03	30.09	[0, 40]
Short Field Takeoff Weight	[kg]	44363.2	44286.1	[20000, 100000]
Design Takeoff Weight	[kg]	45566.0	45565.1	[20000, 100000]
Design Cruise Altitude	[km]	13.38	13.38	[6, 14]
Design Cruise Velocity	[knots]	471.31	471.5	[344.142, 516.213]
Short Field Mission Cruise Altitude	[km]	Constant	10.99	[6, 14]
Short Field Mission Cruise Velocity	[knots]	Constant	445.6	[344.142, 516.213]
Maximum Range Cruise Altitude	[km]	Constant	13.50	[6, 14]
Maximum Range Cruise Velocity	[knots]	Constant	474.6	[344.142, 516.213]
Wing Span	[ft]	118.0	118.0	$\leq$ 118.0
Takeoff Noise	[dB]	92.49	92.5	$\leq$ 92.5
Sideline Noise	[dB]	89.99	90.0	$\leq$ 95.0
Landing Noise	[dB]	92.28	92.28	$\leq$ 99.0
Maximum Zero Fuel Weight Consistency	[-]	2.03e-12	3.798e-12	$\geq$ 0
Design Range Fuel Margin	[-]	-2.27e-10	-2.065e-10	$\geq$ 0
Short Field Fuel Margin	[-]	0.00619	0.00432	$\geq$ 0
Maximum Range Fuel Margin	[-]	-5.39e-10	-8.50e-9	$\geq$ 0
Takeoff Field Length	[m]	1069.4	1061.9	$\leq$ 2056
Short Takeoff Field Length	[m]	888.0	885.4	$\leq$ 1165
Landing Field Length	[m]	1659.4	1659	$\leq$ 1700
Second Segment Climb Gradient Max Range	[-]	0.223	0.225	$\geq$ 0.024
Second Segment Climb Gradient Short Field	[-]	0.260	0.261	$\geq$ 0.024
Maximum Throttle	[-]	0.474	0.499	$\leq$ 1
Design Fuel Burn	[kg]	3139.6	3138.9	-
Function Evaluations	[-]	1216	1811	-

result is better than could be expected in practice since as we mentioned above, our wing weight model does not punish the design for taper equal to zero and a very large sweep value. The cost of incorporating mission variables, however, significantly increased the number of function evaluations to obtain the slightly improved optimum. We have only considered a tube-and-wing configuration in our studies and cruise mission parameters.

Beyond SNOPT, we studied the COBYLA, SLSQP, and ALPSO algorithms with SUAVE. ALPSO works well to find the optimum, but took approximately an order of magnitude more function evaluations for the two dimensional test cases considered compared to the gradient-based methods. SLSQP works well in the SciPy package, but this algorithm is not able to be run in parallel making a problem with large number of design variables intractable. The PyOpt version of SLSQP had some bugs that resulted in solutions that were not optimal. We have not had time to determine what caused this difference in results. The PyOpt version of COBYLA did not find the optimum in any of cases we considered. In addition, we tried to use IPopt in connection with SUAVE to optimize aircraft, but could not even get basic test cases to optimize. Overall, SUAVE can be treated as a black-box function to which external off-the-shelf optimizers can be connected, but these optimizers should also be tested to make sure they treat the problem as expected.

Acknowledgments

The authors would like to thank Tarik H. Orra and Carlos R. Ilario da Silva from Embraer for their help in setting up the baseline transport aircraft. They would also like to thank Michael Vegh for creating the contour function applied to better understand understand the two variable design spaces. Emilio Botero would like to acknowledge the support of the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program.

References

[1]Lukaczyk, T. W., Wendorff, A. D., Colonno, M., Economon, T. D., and Alonso, J. J., "SUAVE: An Open-Source Environment for Multi-Fidelity Conceptual Vehicle Design," *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA 2015-3087, June 2015.

[2]"Embraer 190," Tech. Rep. APM-1901, Embraer S. A., August 2005.

[3]Botero, E., Wendorff, A. D., Lukaczyk, T. W., Alonso, J. J., Orra, T., and Ilario, C., "SUAVE: An Open-Source Environment for Conceptual Vehicle Design and Optimization," *54th AIAA Aerospace Sciences Meeting*, AIAA 2016-1275, January 2016.

[4]Trent Lukaczyk, et al., "SUAVE 0.3.0," May 2016, http://dx.doi.org/10.5281/zenodo.50810.

[5]Perez, R. E., Jansen, P. W., and Martins, J. R. R. A., "pyOpt: A Python-Based Object-Oriented Framework for Nonlinear Constrained Optimization," *Structures and Multidisciplinary Optimization*, Vol. 45, No. 1, January 2012, pp. 101–118.

[6]Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrainted Optimization," *Society for Industrial and Applied Mathematics*, Vol. 47, No. 1, February 2005, pp. 99–131.

[7]Kraft, D., "A Software Package for Sequential Quadratic Programming." Tech. Rep. DFVLR-FB 88-28, DLR German Aerospace Center - Institute for Flight Mechanics, Koln, Germany, 1988.

[8]Powell, M., "A Direct Search Optimization Method that Models the Objective and Constraint Function by Linear Interpolation," *Advances in Optimization and Numerical Analysis*, 1994, pp. 51–67.

[9]Jansen, P. and Perez, R., "Constrained structural design optimization via a parallel augmented Lagrangian particle swarm optimization approach," *Computers & Structures*, Vol. 89, No. 1314, 2011, pp. 1352 – 1366.