Sizing Methods for Aircraft of Variable Propulsion System Complexity

J. Michael Vegh*, Tim MacDonald †, Andrew D. Wendorff‡, Juan J. Alonso§
Stanford University, Stanford, CA, 94305, U.S.A.

This paper compares the computational cost and robustness of a number of different algorithms for the sizing and optimization of aircraft. In particular, three classes of aircraft will be investigated, each with a different level of propulsion system complexity. The first is a conventional turbofan-powered aircraft. A lithium-air battery-powered aircraft, incorporating both energy and power constraints is second. Third is an aluminum-air/lithium-ion battery powered aircraft, which switches to different energy systems depending on the overall propulsion system power requirements. Several different aircraft design problem formulations will be considered for each of these aircraft. Sizing loops as well as determining constraints such as fuel margin by exposing them to the optimizer will both be explored. Results suggest that integrating surrogate models to inform the sizing loop leads to substantially fewer function evaluations than a simple successive substitution, while maintaining a relative insensitivity to a poorly-informed initial guess that optimizer-sizing algorithms lack.

Nomenclature

AoA	angle of attack
AR	aspect ratio
E	energy
Esp	specific energy
f_{aux}	auxiliary power fraction
h	sizing constraint
m	mass
n_{opt}	number of optimizer calls (including finite difference steps)
n_{size}	number of sizing loop calls
P	power
Psp	specific power
R	range
SFC	specific fuel consumptions
$\frac{t}{c}$	thickness to chord ratio
V	velocity magnitude
W	Weight
x	optimization variables
y	sizing variables
WL	wing loading
λ	taper ratio

Subscripts

*Ph.D. Candidate, Department of Aeronautics & Astronautics, AIAA Student Member.
†Ph.D. Candidate, Department of Aeronautics & Astronautics,AIAA Student Member.
‡Ph.D. Candidate, Department of Aeronautics & Astronautics, AIAA Student Member.
§Professor, Department of Aeronautics & Astronautics, AIAA Associate Fellow.

1,2,3...	segment number
i	initial
f	final
j	iteration number
prim	primary
out	parameter returned by the mission solver

I. Introduction

Aircraft conceptual design is an inherently iterative process due to the fact that many of the methods employed do not have an analytical solution.[1] Additional design requirements increase the computational cost of these iterations, especially in the case of unusual aircraft configurations; traditional tube-and-wing designs rely heavily on the use of empirical correlations for initial design performance, while more unusual designs with less data available require the direct use of expensive physics-based tools in order to properly estimate aircraft performance.[2] This makes each aircraft iteration much more computationally expensive; the designer should therefore reduce the number of sizing iterations to as few as possible. Furthermore, for extremely unconventional cases, the designer may have little-to-no intuition as to what a "good" initial guess for the aircraft might look like. While the cost of introducing higher-fidelity analysis may be mitigated through the use of clever MDO architectures, the lower specific energy of battery systems has a cascading effect on the mass, and by extension, structural size of the aircraft.[3,4] Properly evaluating the propulsion system is essential to the sizing process of futuristic electric aircraft, as even generous estimates of technology improvements still falls well short of what one may obtain with fossil fuels.[5] To illustrate some of the difficulties, taking the Breguet Range Equation.

$$R = \frac{V}{SFC} * \frac{L}{D} * ln(\frac{W_i}{W_f}) \tag{1}$$

and rearranging for fixed range

$$\frac{W_i}{W_f} = \exp(\frac{R * SFC}{V} * \frac{1}{\frac{L}{D}}) \tag{2}$$

demonstrates that the required "fuel" weight ($W_i - W_f$) is proportional to the exponential of the effective specific fuel consumption of the aircraft (which specific energy is in turn inversely proportional to). Analogous equations also exist for electric aircraft, but the adverse effect on total weight is more pronounced, due to the fact that overall vehicle weight does not decrease during flight.[6] Even the most optimistic estimates for battery capacity fall well short of fossil fuel specific energy, as seen in Table 1.

Table 1: Energy Source Comparison

source	specific energy (W-h/kg) (approximate)	specific power (kW/kg)
jet fuel	12,000	
lithium-air battery	2,000	.66
aluminum-air battery	1,300	.2
lithium-ion battery	200	1

Table 1 lists the battery configurations from most optimistic (lithium-air, which is an area of active research) to least optimistic (lithium-ion, which is currently used in automobiles). Aluminum-air batteries are currently available with these power/energy characteristics, but have serious limitations such as low specific power and non-rechargability. Even the most optimistic battery choice has 1/6 the specific energy of current propulsion systems, while current widely-available technology has a specific energy 1/60th that of jet fuel. Thus, configuration improvements, such as improving L/D or reducing the structural weight are crucial to implementing these "green" aircraft with any significant amount of range, which may necessitate the use of higher fidelity methods. Furthermore,because this significantly increases the range factor, it also means that

the overall aircraft weight will be more sensitive to design choices, making it more important to "nail down" the configuration early on in the design process. This means the introduction of higher fidelity methods in the conceptual design stage could have a much bigger impact. Therefore, efficiently determining the required size of the vehicle is critical in the development of these futuristic aircraft concepts. The generalized problem to be solved in this paper is as follows:

$$\min f(x) \tag{3}$$

$$s.t. \quad g(x) > 0 \tag{4}$$

$$h(y) > 0 \tag{5}$$

$$h(y) = (y - y_{out})/y \tag{6}$$

$$y > 0 \tag{7}$$

where x is a set of design variables, f is the objective, and y is a set of terms called "sizing variables," which include parameters such as vehicle mass, battery energy, as well as propulsion system power. h(y) is the vector of constraints associated with each y. For a modern commercial aircraft, y could correspond to the gross takeoff weight of the vehicle while h(y) could correspond to a fuel margin constraint. Electric aircraft would include requirements such that the battery state of charge remains positive, and the power requirements of the mission do not exceed the maximum power output of the battery. Note that in the context of this paper, a "sizing evaluation" refers to when vehicle's geometry and weight is defined and runs the mission based on the inputs (x,y). This paper will investigate the computational efficiency and robustness of a number of aircraft-sizing schemes applied to three different aircraft classes. First, a conventional turbofan-powered aircraft will be investigated, with a single sizing variable of y = GTOW. This conventional case will be used as a benchmark, and is most representative of aircraft built today. A drawing of a conventional aircraft using OpenVSP can be seen below.[7]

Figure 1: Conventional Turbofan-Powered Aircraft

The second aircraft is a lithium-air battery powered design; details of the analysis and design procedure can be seen in reference 8. This particular battery chemistry has an estimated specific energy of 2000 W-h/kg, with a specific power of .66 kW/kg. Battery technology here is considered futuristic, indicative of what one might call a "best case scenario." The sizing variables for this configuration can be seen below, along with a CAD drawing of an example aircraft (with a box representing the required lithium-air battery volume) in Figure 2.

$$y = [m, E_{prim}, P] \tag{8}$$

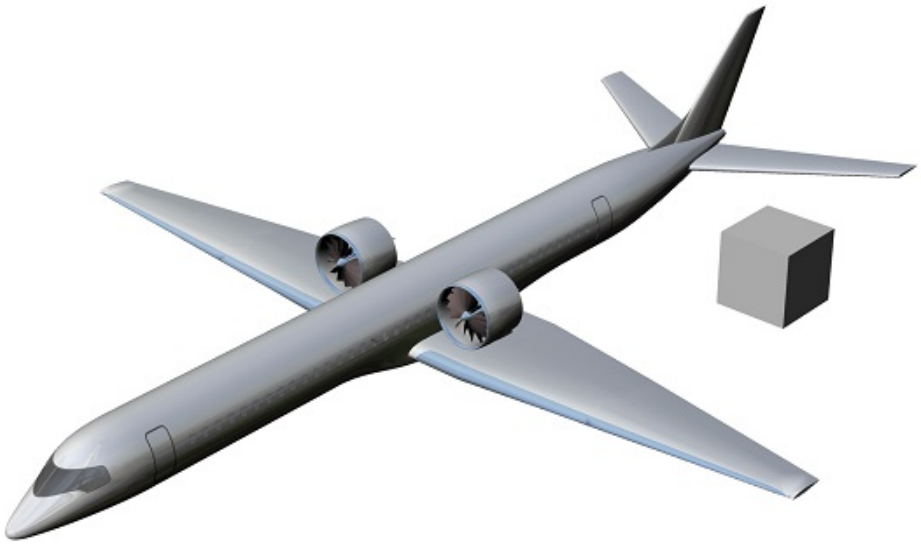

Figure 2: Lithium-Air Aircraft

The most-complicated case is an aluminum-air powered aircraft, augmented with a lithium-ion battery for high-power operation. A detailed breakdown of the propulsion system characteristics can seen in reference 9; in summary, the aircraft is powered by an aluminum-air battery which has a high specific energy (1300 W-h/kg), but low specific power, along with a lithium-ion battery (which has a higher specific power, but much lower specific energy). The lithium-ion battery meets a specified fraction of the power requirements of the vehicle, and activates whenever the maximum power output of the aluminum-air battery is exceeded. The sizing variable vector y for this configuration can be seen below.

$$y = [m, E_{prim}, E_{aux}, P] \tag{9}$$

Technology used is representative of batteries available today, with the caveat that current technology does not allow for recharging the aluminum-air battery, although the models include estimated costs for recycling the battery after discharge. The aluminum-air design possesses the most complex propulsion-system investigated here, and is thought to be the most representative of the power/energy tradeoffs of next-generation hybrid-electric vehicles. A CAD image of an example aircraft from the previous paper can be seen in Figure 3(which includes the required volumes of the energy subsystems within the wing, where blue is water, red is a lithium-ion battery, and yellow is an aluminum-air battery).[9]

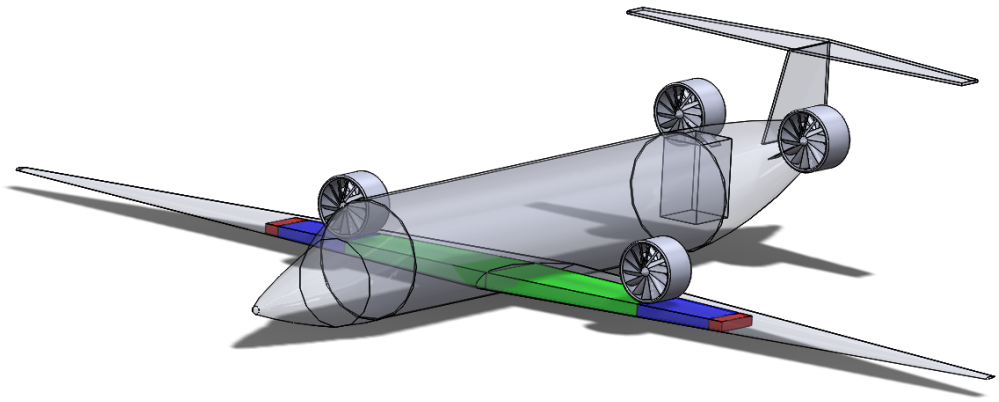

Figure 3: Aluminum-Air/Lithium Ion Powered Aircraft

Section II details the different sizing methods employed in this study, while Section III contains a summary comparing their computational cost (in terms of number of function evaluations), using a variety of different algorithms for a full optimization case.

II. Methodology

All of the aircraft analysis and design/optimization studies will utilize SUAVE, an open-source design tool.[10] SUAVE was built to ensure compatibility with any number of exotic energy systems, and thus, is well-suited for studies such as this. A more recent paper highlighted SUAVE's flexibility in formulating optimization problems which this paper will heavily leverage.[11] All optimization cases were handled using SUAVE's pyOpt optimizer wrapper, which in turn called SNOPT (a Sequential Quadratic Programming method).[12,13] This paper uses two different aircraft sizing procedures, comparing their robustness and speed within an optimization problem for the three different aircraft classes.

One sizing procedure is termed "optimizer-sizing" within the context of this paper. The algorithm can be explained as follows; the sizing variables are appended to the optimization problem with the relevant constraints handled by the optimizer to ensure aircraft consistency, i.e. equation 3 is replaced with equation 10, while equation 5 is handled by the optimizer as shown in equations 10-14.

$$min f(x,y) \tag{10}$$

$$s.t. \quad g(x) > 0 \tag{11}$$

$$h(y) > 0 \tag{12}$$

$$h(y) = (y - y_{out})/y \tag{13}$$

$$y > 0 \tag{14}$$

This approach tends to be favored by optimization algorithm developers, as it is often more efficient to allow the optimizer to search along an infeasible region, rather than solve a large number of subproblems.[14] One disadvantage is that, in the event that the optimization fails to converge to a feasible point, the designer gains no new knowledge of the design space. Additionally, it was found in a previous paper that, for hybrid-electric aircraft, formulating the problem in this manner caused difficulties in satisfying the sizing constraints.[15] As the problem becomes more complex, the sizing variables may become loosely coupled, and an abundance of local minima arises, causing difficulty even for generally reliable optimizers. Nonetheless, this paper explores this "optimizer sizing" algorithm, comparing its effectiveness (in terms of both robustness as well as computational cost, in number of function evaluations) as the propulsion system complexity changes. Another formulation is to solve the subproblem for y as a function of x, as shown in equations 15-18.

$$\min f(x) \tag{15}$$

$$s.t. \quad g(x) > 0 \tag{16}$$

$$\text{solve for y} \quad \text{s.t.} \quad -\epsilon < h(y) < \epsilon \tag{17}$$

$$h(y) = (y - y_{out})/y \tag{18}$$

This "sizing-loop" method is the general formulation used by most aircraft manufacturers as well as students in conceptual design. It has a number of advantages, in that, in the problems considered here, any optimal configuration would have an h(y) = 0. Thus, the optimizer may be less likely to become "stuck" in local minima (at least those defined by the sizing variables). Furthermore, as each optimizer iteration is properly sized, the designer can build some intuition about the sensitivity of the aircraft to input variables as the optimization problem progresses. A major disadvantage is a potential loss in solution accuracy; the solution can only be converged to a certain value, practically speaking, before accumulated numerical errors prevent further convergence (for the aluminum-air aircraft, this was found to begin occurring at a sizing tolerance of 1E-5). This means that the finite differencing steps from the optimizer should be increased to ensure that they are outside of the optimizer bounds; the users guide to SNOPT recommends a step size $\approx (tol)^{1/2}$.[13] A basic information tree using these sizing variables can be seen in Figure 4 below. Note that this particular algorithm is called at every optimizer iteration.

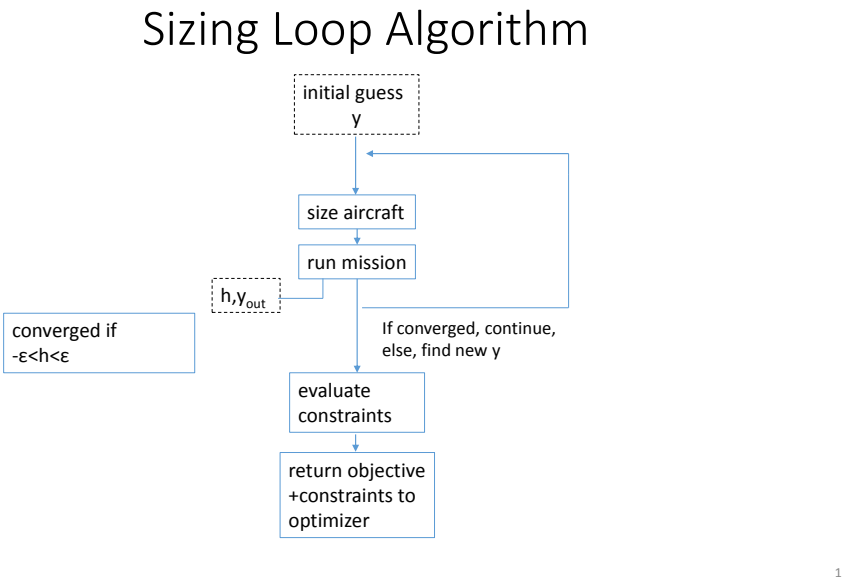

Figure 4: Sizing-Loop Algorithm

A common method used to solve this inner-loop problem is successive substitution, where y_{out} is set as a new y, and the loop iterated on until the guesses converge. This is a fairly slow, but robust method. Solving this problem is relatively simple for a single variable, but becomes more complicated as more sizing variables are added especially when they are all loosely coupled. For example, for the aluminum-air aircraft case, the auxiliary battery often only consumes a small amount of the total energy, yet accounts for a substantial fraction of the power; thus, using total energy as a sizing variable results in highly infeasible aircraft, and needs to be handled separately. To illustrate, Figure 5 compares the value of the sizing constraints at each successive substitution iteration

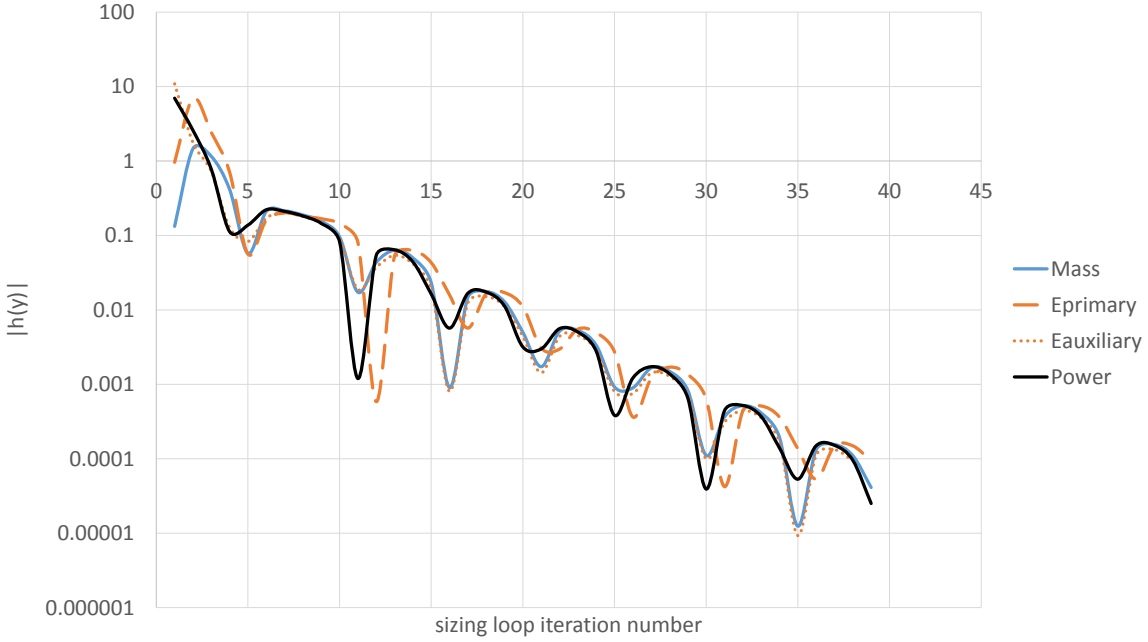

Figure 5: Convergence Results for Higher-Dimensional y

Figure 5 corresponds to an aluminum-air/lithium-ion battery powered aircraft. As the solution proceeded (for fixed x), there was significant oscillation in the error; furthermore, the different sizing errors were out of phase with each other, which significantly slowed the convergence rate Furthermore, to converge all of the required parameters, some of the other parameters may become over-converged, which, due to the higher range factor, may affect the design enough to result in a serious reduction in gradient accuracy. One method to reduce the number of sizing iterations is to use Newton-Raphson (NR), where the algorithm in Figure 4 is finite-differenced in y to obtain the Jacobian, and the new y value becomes

$$y_{j+1} = y_j - (J_j^{-1}) * h(y_j) \tag{19}$$

Newton-Raphson is known to exhibit quadratic convergence near the solution, which can significantly reduce the number of function evaluations. Thus, NR methods in the paper will use successive substitution until h(y) is converged to within 5 % for all values of h, then use NR to accelerate convergence. Additionally, depending on the sizing tolerance, newton-raphson, due to its quadratic convergence, may cause a particular sizing iteration to become overconverged, introducing numerical noise to the outer loop optimizer. Another method to increase code speed is to use Broyden's method with the Sherman-Morrison formula to directly update the inverse of the Jacobian.[16]

$$J_{j+1}^{-1} = J_j^{-1} + \frac{dy - J_j^{-1} dh dy^T}{dy^T J_j^{-1} dh} dy^T J_j^{-1} \tag{20}$$

where

$$dy = y_{j+1} - y_j \tag{21}$$

and

$$dh = h(y_{j+1}) - h(y_j) \tag{22}$$

This reduces the number of function evaluations by eliminating the need to need to finite difference to form the Jacobian at each iteration (which can become expensive as the dimension of y increases). However, this update is also less robust, and the Jacobian may need to be reinitialized to ensure convergence. A demonstration of these methods "in action" can be seen in Figure 6, which compares the mean squared error of h at each sizing loop iteration for the aluminum-air reference guess for successive substitution, Newton-Raphson, and Broyden's Methods. Gaps in the iterations correspond to finite-difference steps to build the Jacobian

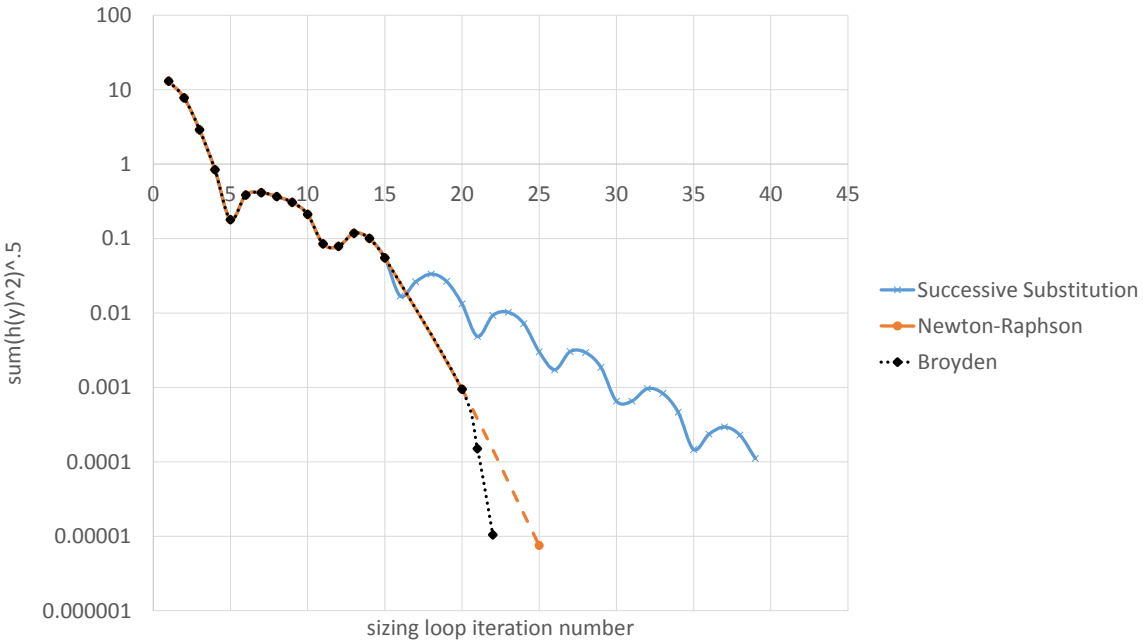

Figure 6: Convergence Results for higher-dimensional y

Figure 6 shows that, using these methods can solve the sizing loop problem considerably faster than simple successive substitution. However, the user should be careful, in that, due to their quadratic convergence, they may overshoot the solution tolerance (which here is 1E-4), which may cause significant problems when run with gradient-based optimizers. In this instance, the convergence jumped from 1E-3 to 1E-5 in a single iteration. This is a less critical issue when one uses global-optimization techniques, as accurate gradients are not nearly as important there. Figure 5 also shows that properly informing the initial guess for y is a crucial aspect of increasing code speed. To that end, a number of different techniques were used to reduce the number of sizing loop iterations, thereby improving the optimization efficiency. One easy way to increase code speed (which is effective for gradient-based optimizers), is to tabulate the converged solutions, and, when the input variables x are "close" to a tabulated value use the nearest converged y(x) as an initial guess. However, farther away from the tabulated data, one should use other methods to determine a "good" initial guess for the y values. To that end, a variety of regression algorithms from scikit-learn were evaluated to determine their effectiveness in finding a good guess for the initial y in the sizing loop.[17] Details can be seen in the Results section of this report.

III. Results

Because the aluminum-air aircraft was the most complicated case, training data from a full optimization case was generated and used to evaluate several regression algorithms to determine a "good" initial guess.

This aircraft has 4 sizing variables as seen in equation 9, along with twelve input variables shown in Table 2. Note that each of the aircraft cases in the report uses the same design variables (aside from f_{aux}, which is unique to the aluminum-air aircraft), although their bounds differ from case to case.

Table 2: Aluminum-Air Design Variables

variable	Lower Bound	Upper bound
f_{aux}	0	1.
Wing Loading $(\frac{kg}{m^2})$	200	800
Thrust Loading	.05	.6
Fan Pressure Ratio	1.1	2.4
$\frac{t}{c}$	.07	.2
$V_{climb_1}(\frac{m}{s})$	50	140
$V_{climb_2}(\frac{m}{s})$	50	140
$V_{climb_3}(\frac{m}{s})$	50	140
$h_{cruise}(ft.)$	20,000	30,000
$h_{frac,climb_1}$	.1	1
$h_{frac,climb_2}$	.2	1
$h_{frac,descent_1}$	.1	1

As the optimization process proceeded, the algorithm was trained on iteration 1 through i, and tested on iteration i+1, where the norm of the error was output. In other words

$$\begin{aligned} &\text{for j in converged results} \\ &\quad \text{train on 1:j-1} \\ &\quad \text{test on j} \end{aligned} \tag{23}$$

Some statistics were then calculated to compare the effectiveness of each algorithm over the entire dataset. Table 3 compares some representative linear-regression algorithms (many others were tested, but these proved to be the most effective on this set of data). Note that, the mean, median, and maximum errors were the most significant metrics for algorithm performance. The mean and median both represent the performance the algorithm is most likely to have when evaluating a particular optimization point, while the maximum error, if too high (meaning a poor initial guess), could cause the sizing loop to diverge, which could result in a breakdown of the optimization process (when run in-the-loop).

Table 3: Aluminum-Air Aircraft Outer Loop Performance (Linear Regression Algorithms)

algorithm	mean err	median err	std err	min err	max err
SVR (RBF Kernel)	0.830	0.432	1.576	0.070	6.935
Gaussian Process	0.462	0.300	0.433	0.025	2.297
Table	0.784	0.062	2.183	0.001	11.166
Kneighbors (5), distance weighted	0.454	0.131	1.037	0.006	6.505

In addition, several ensemble methods were tested on this dataset. Note that, the ensemble methods are stochastic, so the results will vary slightly each time the method is run. However, after running each several times, it was found that the results in Table 4 were representative of the overall performance of each method.

Table 4: Aluminum-Air Aircraft Outer Loop Performance (Ensemble Algorithms)

algorithm	mean err	median err	std err	min err	max err
Random Forest	0.403	0.293	0.402	0.003	2.110
Gradient Boosting	0.324	0.147	0.395	0.002	1.898
Bagging Regressor	0.400	0.361	0.366	0.008	1.721
Extra Trees Regressor	0.316	0.137	0.380	0.009	1.901

For these cases, the simple Table lookup model possessed the lowest median error, but a high maximum error (and with it, a higher standard deviation), then the other methods. The max error occurs when the optimizer takes large steps, which tends to happen early on in the optimization process. Compared to the table-lookup method, K Nearest Neighbors (KNN) has lower maximum, mean, and median errors, than the Table-Lookup method. Gaussian Process Regression (GPR) reduced the maximum error by a factor of three, but with lower average accuracy. This is indicative of a more robust, but less accurate algorithm. The ensemble methods, on the other hand, all appear to maintain the robustness of GPR, with the addition of increased accuracy, with Extra Trees edging the other algorithms out in terms of performance. With the algorithm performance estimated for the most-complicated case, the table-lookup, K Nearest Neighbors (chosen to have distance-weighted coefficients, with 5 neighbors), Gaussian Process Regression, and Extra Trees were evaluated and compared for the three aircraft classes. Each of these three aircraft classes were also evaluated for the three inner-loop methods (successive-substitution, newton-raphson, and broyden), as well as optimizer sizing. Sizing Loop results for the regional jet case are shown in Tables 5-7. Note that, because of numerical noise (sizing was converged to h=1E-4), each of these algorithms may have resulted in a different number of outer loop iterations. Thus, $\frac{n_{size}}{n_{opt}}$ is considered the best metric for evaluating algorithm performance, as it is the average number of sizing calls per optimization iteration.

Table 5: Regional Jet Optimization Results (Successive Substitution)

loop evaluation	initial step	n_{size}	n_{opt}	$\frac{n_{size}}{n_{opt}}$	landing weight (lbs.)
Successive Substitution	Same Point	406	57	8.8	83,282
Successive Substitution	Table	149	46	3.2	83,454
Successive Substitution	KNN(5)	150	46	3.3	83,454
Successive Substitution	GPR	150	46	3.3	83,454
Successive Substitution	Extra Trees	149	46	3.2	83,457

Table 6: Regional Jet Optimization Results(Newton-Raphson)

loop evaluation	initial step	n_{size}	n_{opt}	$\frac{n_{size}}{n_{opt}}$	landing weight (lbs.)
Newton-Raphson	Same Point	312	145	6.8	83,259
Newton-Raphson	Table	177	46	3.8	83,252
Newton-Raphson	KNN(5)	238	46	5.2	83,274
Newton-Raphson	GPR	220	46	4.8	83,267
Newton-Raphson	Extra Trees	229	46	5.0	83,271

The regional jet case was interesting, in that, the problem was simple enough that the optimization case converged after only ≈ 5 major iterations, meaning that the regression algorithms only ran for one or two iterations (three major iterations was chosen as the threshhold before these regression were run, to prevent extrapolation to extremely unphysical results). Thus, the extra complication in setting up the more elaborate algorithms may not necessarily be cost-effective in a more applied setting. Additionally, the more exotic

Table 7: Regional Jet Optimization Results(Broyden)

loop evaluation	initial step	n_{size}	n_{opt}	$\frac{n_{size}}{n_{opt}}$	landing weight (lbs.)
Broyden's Method	Same Point	312	46	6.8	83,261
Broyden's Method	Table	211	57	3.7	83,255
Broyden's Method	KNN	205	57	3.6	83,249
Broyden's Method	GPR	196	57	3.4	83,251
Broyden's Method	Extra Trees	269	68	4.0	83,321

loop-evaluation methods (Newton-Raphson and Broyden) proved to be somewhat more computationally expensive; this is because they had a tendency to "overshoot" the solution when SNOPT performed the finite difference iterations, resulting in less accurate gradients. Furthermore, because most of the sizing loop iterations were taken during optimizer finite difference steps, the initial guess tended to be "near" the solution value (for the regression methods). Thus the extra sizing loop iterations to construct the Jacobian may actually be slower than successive substitution for a given step. Table 8 shows the results for when the y vector (in this case, GTOW) is moved to the outer loop optimization, with h added as a constraint. The uninformed initial guess uses the default GTOW from the sizing loop problem. Informed optimizer sizing uses the sized GTOW from iteration 1 of the sizing loop problem as the initial GTOW in the optimization problem.

Table 8: Regional Jet Optimization Results (Optimizer Sizing)

loop evaluation	initial step	n_{size}	n_{opt}	$\frac{n_{size}}{n_{opt}}$	landing weight (lbs.)
uninformed optimizer sizing	N/A	254	254	1	82,513
informed optimizer sizing	N/A	350	350	1	82,513

In this particular case, the uninformed case converged more quickly, as the optimizer used steepest descent to rapidly move through the infeasible region. The informed case, on the other hand, tended to hug the fuel margin constraint line, increasing the number of optimizer iterations, but converges to the same result. Tables 9- 11 show results for the more complicated lithium-air-powered aircraft.

Table 9: Lithium-Air Jet Optimization Results (Successive Substitution)

loop evaluation	initial step	n_{size}	n_{opt}	$\frac{n_{size}}{n_{opt}}$	landing weight (lbs.)
Successive Substitution	Same Point	3,112	326	9.55	131,272
Successive Substitution	Table	192	62	3.1	134,208
Successive Substitution	KNN(5)	171	50	3.4	134,277
Successive Substitution	GPR	206	62	3.3	134,443
Successive Substitution	Extra Trees	197	62	3.1	134,231

Table 10: Lithium-Air Jet Optimization Results(Newton-Raphson)

loop evaluation	initial step	n_{size}	n_{opt}	$\frac{n_{size}}{n_{opt}}$	landing weight (lbs.)
Newton-Raphson	Same Point	3,894	350	11.1	130,918
Newton-Raphson	Table	765	134	5.7	131,210
Newton-Raphson	KNN(5)	2,632	640	4.1	133,056
Newton-Raphson	GPR	1,052	218	4.8	130,944
Newton-Raphson	Extra Trees	3,415	74	5.7	132,951

Table 11: Lithium-Air Jet Optimization Results Summary(Broyden)

loop evaluation	initial step	n_{size}	n_{opt}	$\frac{n_{size}}{n_{opt}}$	landing weight (lbs.)
Broyden's Method	Same Point	2,845	314	6.8	131,371
Broyden's Method	Table	867	182	3.7	131,101
Broyden's Method	KNN	422	112	3.6	133,419
Broyden's Method	GPR	214	62	3.4	133,715
Broyden's Method	Extra Trees	714	170	4.0	130,969

The lithium-air case demonstrates the "overshooting" problem with Newton-Raphson, as it took a substantially larger number of optimization problem calls to converge than the other cases. Additionally, from inspection, none of the initial step methods (Table, KNN,GPR, or Extra Trees) appears to perform significantly better than the others. Note that, like the regional jet case, the lithium-air jet optimization problem was found to converge within only a few optimizer iterations (at least for successive substitution), so the effectiveness of each algorithm is difficult to compare. Nonetheless, for successive substitution, the overall number of sizing calls was lower for each of these methods by an order of magnitude than by starting from the same initial y values. Broyden's method mitigates some of the numerical noise issues seen in Newton-Raphson, but still appears to be marginally outperformed by the successive substitution methods (on average). Optimizer sizing results can be seen in Table 16.

Table 12: Lithium-Air Jet Optimization Results (Optimizer Sizing)

loop evaluation	initial step	n_{size}	n_{opt}	$\frac{n_{size}}{n_{opt}}$	landing weight (lbs.)
uninformed optimizer sizing	N/A	2,642	2,642	1	137,735
informed optimizer sizing	N/A	1,262	1262	1	136,015

For this more complicated case, the sizing-loop methods, for the most part, tend to outperform the optimizer-sizing architecture. The optimizer was found to have some difficulty determining the sizing constraints, as energy, power, and weight are all coupled in an aircraft; therefore, solving the subproblem was found to be more efficient. Furthermore, a fully-sized initial guess resulted in a faster optimization process, as the coupling of the sizing variables proved troublesome for the optimizer. Finally, results for the aluminum-air aircraft can be seen in Tables 13-15.

Table 13: Aluminum-Air Optimization Results(Successive Substitution)

loop evaluation	initial step	n_{size}	n_{opt}	$\frac{n_{size}}{n_{opt}}$	landing weight (lbs.)
Successive Substitution	Same Point	95,849	4,396	21.8	44,976
Successive Substitution	Table	7,227	1,343	5.4	45,040
Successive Substitution	KNN(5)	11,838	2,409	4.9	45,134
Successive Substitution	GPR	3,415	524	6.5	48,454
Successive Substitution	Extra Trees	10,555	2,045	5.2	44,979

Table 14: Aluminum-Air Optimization Results(Newton-Raphson)

loop evaluation	initial step	n_{size}	n_{opt}	$\frac{n_{size}}{n_{opt}}$	landing weight (lbs.)
Newton-Raphson	Same Point	11,678	433	27.0	48,221
Newton-Raphson	Table	17,398	2,357	7.4	45,412
Newton-Raphson	KNN	11,208	1,695	6.6	48,330
Newton-Raphson	GPR	3,652	550	6.6	47,309
Newton-Raphson	Extra Trees	28,353	4,580	6.2	45,016

Table 15: Aluminum-Air Optimization Results(Broyden)

loop evaluation	initial step	n_{size}	n_{opt}	$\frac{n_{size}}{n_{opt}}$	landing weight (lbs.)
Broyden's Method	Same Point	10,562	328	32.2	45,754
Broyden's Method	Table	14,169	3,194	4.4	44,669
Broyden's Method	KNN	1,582	225	7.0	48,504
Broyden's Method	GPR	3,292	342	9.6	50,460
Broyden's Method	Extra Trees	1,620	173	9.4	50,435

Table 16: Aluminum-Air Jet Optimization Results (Optimizer Sizing)

loop evaluation	initial step	n_{size}	n_{opt}	$\frac{n_{size}}{n_{opt}}$	landing weight (lbs.)
uninformed optimizer sizing	N/A	11,868	11,868	1	66,571
informed optimizer sizing	N/A	5,714	5,714	1	43,804

The aluminum-air aircraft case, like the lithium-air aircraft, had an order of magnitude reduced computational cost when not starting from the same point. Additionally, solving the sub-problem proved more troublesome for the inner-loop solver, due to the increased complexity (as seen in Figure 6). Furthermore, numerical noise became a more significant issue here, which substantially increased optimization time compared to the lithium-air case; note that for each of these algorithms, the optimizer design variables moved very close to the solution output (generally at $\approx$ 1,000 optimization calls (which included finite difference steps)), but was unable to satisfy the optimality conditions, so it spent a significant amount of time searching for better points. Furthermore, for some of the cases, Broyden's method struggled to fully converge, (due to accumulated numerical error), which significantly increased the number of sizing-loop iterations. This may be mitigated by using finite differencing to reinitialize the Jacobian.

The sizing-loop algorithms tended to be the most robust, in that, unless the optimizer-sizing algorithm was started with a "good" initial guess, the optimizer was sometimes unable to converge to a feasible solution

(or at least one "better" than an initial sized result). However, when starting from a fully-sized guess, the optimizer-sizing algorithm was usually the fastest approach, and yielded the best results. Nonetheless, determining a good initial guess for y may be quite complicated, especially when y contains more than 2 parameters. As a result, sizing-loop methods were preferred for these types of problems, as the optimizer had trouble solving the consistency subproblem without a feasible initial guess. Note that, in some cases, it was difficult to consistently compare the effectiveness of determining a good initial guess for the outer loop problem, as the numerical error in the sizing loop process sometimes caused the optimizer to "wander" in certain regions of the design space. A summary plot of the various regression methods using successive substitution can be seen in Figure 7, along with the optimizer sizing results. The "Same Point" step is omitted, to see the spread more clearly. Figure 8 shows the final optimum aircraft weights plotted vs. the ratio of sizing evaluations to optimizer iterations, to more easily compare their performance.

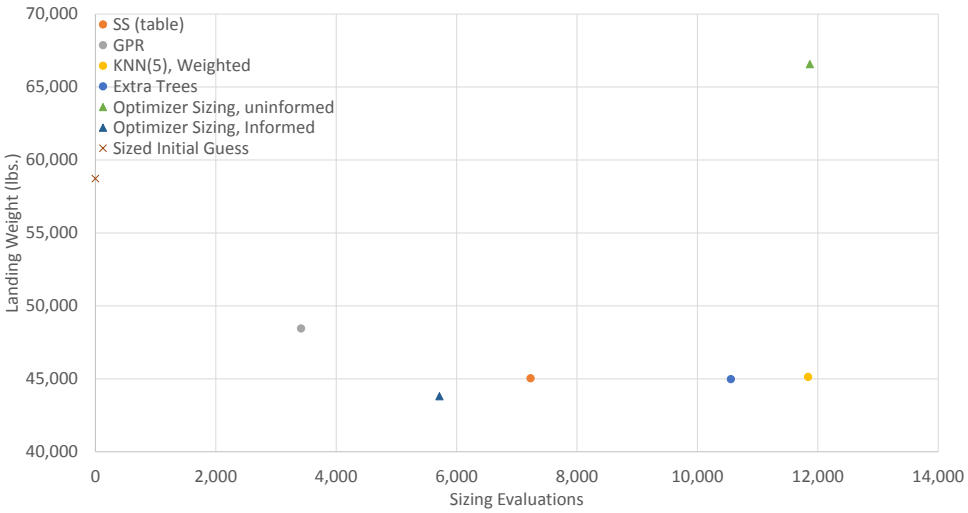

Figure 7: Aluminum-Air Results Summary

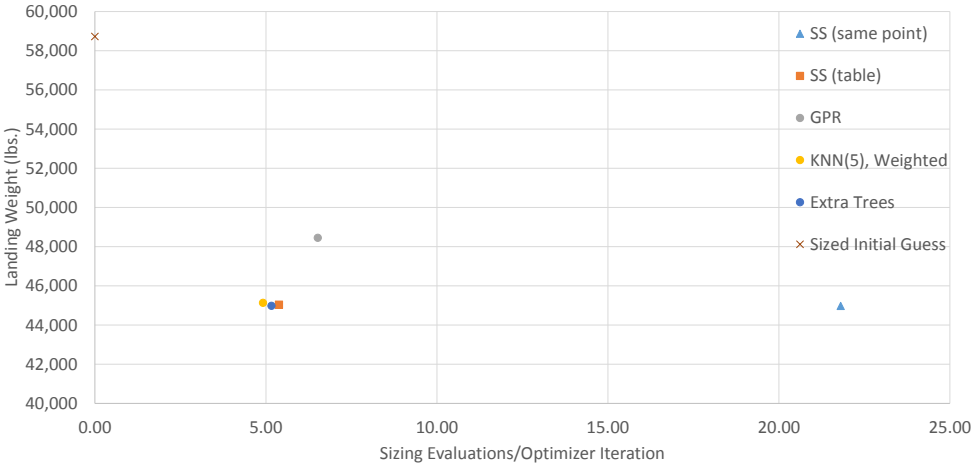

Figure 8: Aluminum-Air Results Summary

In summary, for gas-powered aircraft, ptimizer convergence vs. the use of sizing loops appears to make

relatively little difference in computation time. For aircraft with a single battery (with power and energy requirements), using an informed initial guess within a sizing loop can save an order of magnitude in computational cost over both an uninformed guess as well as using an optimizer. On the other hand, for aircraft with multiple energy systems, there is an extra order of magnitude in computational cost due to the difficulty in solving the subproblem. To illustrate, a plot of the convergence for the three aircraft cases can be seen in Figure 9

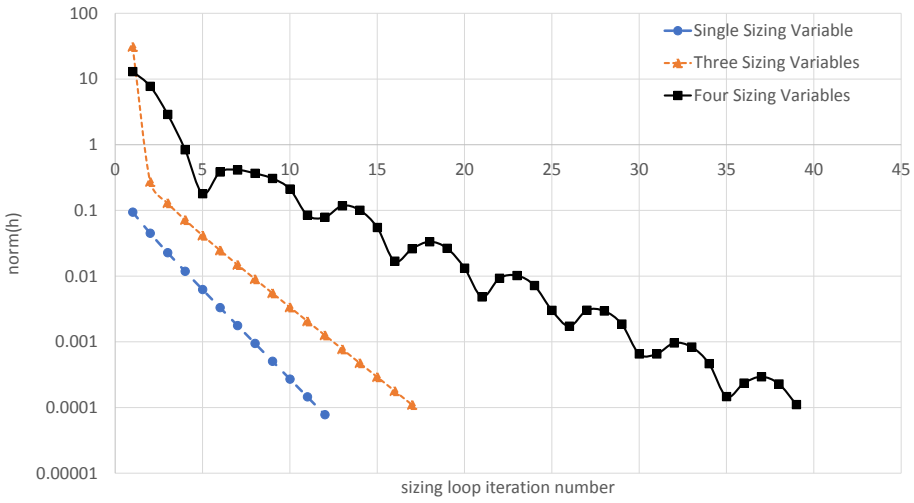

Figure 9: Convergence vs. Number of Sizing Parameters

Both a single sizing variable (indicating a fossil fuel-powered aircraft), as well as a three sizing variable (indicating a battery-powered aircraft) show log-linear convergence when using successive substitution. Including an additional battery system causes significant oscillation in convergence, greatly increasing computational time. This is because, when one energy system requires more energy at a given iteration, the other requires less, leading to a back-and-forth oscillatory impact, where convergence in one system hurts convergence in the other. Using the optimizer to converge this case is somewhat more effective when starting from a well-posed initial guess; however, determining a good initial guess may be nontrivial. In all cases, this leads to an additional order of magnitude computational time over the single energy system case.

IV. Conclusions

Results show a number of trends for these methods. Firstly, that sizing-loop methods appear to be less sensitive to an uninformed initial guess than optimizer sizing, indicating that, solving the sub-problem within each major iteration may be a more robust way of solving an aircraft design/optimization problem. However, due to computational convergence limits, they must be run to a lower sizing tolerance, making them more prone to becoming stuck in local minima. Starting the optimization case with a fully sized vehicle appears to result in a lower computational cost and is less sensitive to local minima. Using surrogate methods to inform the initial guess within the sizing loop methods appears to recover much the code speed of the optimizer-sizing cases, while maintaining robustness. Future work will explore some of these methods using global optimization methods, along with further investigation of inner-loop zero-finding methods.

Acknowledgments

Michael Vegh would like to acknowledge the support of the DoD through the Science Mathematics and Research for Transformation (SMART) Scholarship Program.

Timothy MacDonald would like to acknowledge the support of the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program.

References

[1]Raymer, D., *Aircraft Design: A Conceptual Approach*, AIAA, Playa del Ray, California, 4th ed., 2006.

[2]Fredericks, W., Moore, M., and Busan, R., "Benefits of Hybrid-Electric Propulsion to Achieve 4x Increase in Cruise Efficiency for a VTOL Aircraft," AIAA Aviation Technology, Integration, and Operations (ATIO) Conference.

[3]Martins, J. and Lambe, R., "Multidisciplinary Design Optimization: A Survey of Architectures," *AIAA Journal*, Vol. 51, No. 9, 2013, pp. 2049–2075.

[4]Sinsay, J., Tracey, B., Alonso, J., Kontinos, D., Melton, J., and Grabbe, S., "Air Vehicle Design and Technology Considerations for an Electric VTOL Metro-Regional Public Transportation System," *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA, Indianapolis, IN, 2012.

[5]Moore, M. and Fredericks, B., "Misconceptions of Electric Propulsion Aircraft and their Emergent Aviation Markets," *AIAA SciTech*, NASA Langley Research Center, National Harbor, Maryland, 2014.

[6]Hepperle, M., "Electric Flight- Potential and Limitations," German Aerospace Center, 2012.

[7]Fredericks, W., "Aircraft Conceptual Design Using Vehicle Sketch Pad," *AIAA Scitech*, Orlando, FL, January 2010.

[8]Vegh, J., Alonso, J., Orra, T., and Ilario da Silva, C., "Flight Path and Wing Optimization of Lithium-Air Battery Powered Passenger Aircraft," *AIAA 2015-1674 53rd AIAA Aerospace Sciences Meeting*, AIAA Scitech, Kissimee, FL, 2015.

[9]Vegh, J. and Alonso, J., "Design and Optimization of Short-Range Aluminum-Air Powered Aircraft," *54th AIAA Aerospace Sciences Meeting*, AIAA Scitech, San Diego, CA, 2016.

[10]Lukaczyk, T., Wendorff, A., Botero, E., MacDonald, T., Momose, T., Varyar, A., Vegh, J., Colonno, M., Orra, T., Illaria da Silva, C., and Alonso, J., "SUAVE: An Open-Source Environment for Multi-Fidelity Conceptual Vehicle Design," AIAA Aviation Forum 2015, Dallas, TX, 2015.

[11]Botero, E., Wendorff, A. D., MacDonald, T., Variyar, A., Vegh, J. M., Alonso, J. J., Orra, T. H., and Ilario da Silva, C., ""SUAVE: An Open-Source Environment for Conceptual Vehicle Design and Optimization," *AIAA Scitech*, San Diego, CA, January 2016.

[12]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, 2012, pp. 101–118.

[13]Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Journal on Optimization*, Vol. 47, No. 1, 2002, pp. 99–131.

[14]Kroo, I., Altus, S., Braun, R., Gage, P., and Sobieski, I., "Multidisciplinary Optimization Methods for Aircraft Preliminary Design," AIAA Paper -94-4325-CP, Hampton, VA, January 1994.

[15]Vegh, J., Alonso, J., and Sinsay, J., "Modeling of Diesel and Diesel-Electric Hybrid Propulsion Systems for Conceptual Design of Rotorcraft," *Technical Meeting on Aeromechanics Design for Vertical Lift*, American Helicopter Society, San Francisco, CA, 2016.

[16]Sherman, J. and Morrison, W., "Adjustment of an Inverse Matrix Corresponding to Changes in the Elements of a Given Column or a Given Row of the Original Matrix," .

[17]Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, Vol. 12, 2011, pp. 2825–2830.