High Dimensional, Nonlinear, Non-Convex Optimization Problems in the Area of Aircraft and Vehicle Design

Michael Case[1], Jeffrey Haack[2], MoonChang Kim[3], Mandar Kulkarni[4], Darin Mohr[5], Helmi Temimi[6]

Mentor:
John R. Hoffman, PhD
Lockheed Martin Tactical Defense Systems

Abstract

The preliminary design space exploration for Unmanned Aerial Vehicles (UAV) is a difficult and time-consuming task. Despite an abundance of good UAVs in the market, the models used for the design space exploration are highly proprietory. We present a simplistic aerodynamic model for the design of UAV and techniques with potential for (near) real-time exploration of the design space. Another objective of this study is investigation of FLOPS, the open-source Flight Optimization System software package developed at NASA for the exploration of the design space.

1 Introduction and Motivation

Consider the scenario of a customer requesting from an aircraft manufacturer a vehicle where cost is not a consideration. The odds are the company will settle on a design that is correlated to the designs they have made in the past. However, new innovative designs that are significantly better for the customer's mission may be missed. It could then only be a short time before a significantly better design is produced and the company becomes a victim of its past successes.

The aforementioned scenario illustrates the motivation for a pre-conceptual design. If aircraft designers could better visualize the design space then the potential for a design in the neighborhood of the actual optimal is significantly increased. To better enable an aircraft designer in the pre-conceptual design phase we illustrate how one might construct a low-fidelity aerodynamic model. The design space can then be explored for the optima of a class of Non-linear Programming (NLP)/Mixed Integer Non-linear Programming (MINLP) problems. In this work BONMIN, a mixed intger non-linear program solver, is tested on our model and compared to FLOPS,an existing pre-conceptual design tool. If the equations/constraints in the model are "simple enough" as is the case with our low-fidelity model, one can solve these equations numerically for charecterization of the design space. The intended result of the framework is to allow a UAV designer to allow a near real time look at the design space that will enable the designer examine the "sweetspots" of mission performance with respect to the design space.

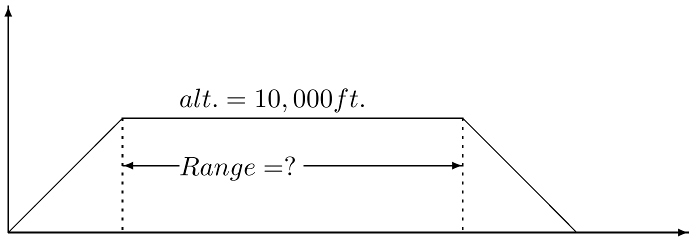
2 Mathematical Framework

This section gives a detailed outline of the model for use in the pre-conceptual UAV design process.

[1]Clemson University
[2]University of Wisconsin
[3]Seoul National University
[4]University of Alabama at Birmingham
[5]University of Iowa
[6]Virginia Polytechnic Institute and State University

2.1 Model

The UAV under consideration is assumed to have the following flight profile.



Flight Profile of the UAV

Following table gives constants, free variables and performance variables in the model. Appendix gives a glossary of the aerodyanamical terms.

Constants	**Free Variables**	**Performance Variables**
Altitude $(h) = 10,000\,ft.$	Gross Take Off Weight (W_0 $lb.$)	Range (R $fts.$)
Skin friction coefficient $(c_{fe}) = 5 \times 10^{-3}$	Engine Horse Power (bhp $hp.$)	Stall speed (V_{stall} $kts.$)
Propeller efficiency $(\eta_p) = 0.75$	Ref. Aerodynamic Area (S_{ref} ft^2)	Endurance (E $hrs.$)
Max. Lift Coefficient $(C_{L_{max}}) = 1.4$	Wetted Area (S_{wet} ft^2)	Runway length (R_L $fts.$)
Empty weight fraction coefficent $(a) = 1.15$	Wing Span (b ft)	Max. Speed (V_{max} $kts.$)
Payload weight $(W_{payload}) = 100\ lb.$		
Gas const. for air $(R_{air}) = 0.286\ KJ/kg/K$		

2.2 Preliminary Calculations

2.2.1 Atmospheric density calculation

For altitudes less than $36152ft.$, atmospheric temperature T in $^\circ F$ and pressure p in lb/ft^2 is given [3] by:

$$\begin{aligned} T &= 59 - 0.00356h \\ p &= 2116\left[\frac{T+459.7}{518.6}\right]^{5.256} \end{aligned}$$

Thus, temperature T_K in $^\circ K$ and pressure p_{SI} in kN/m^2 is

$$\begin{aligned} T_K &= (T + 459.67) \div 1.8 \\ p_{SI} &= 0.047p \end{aligned}$$

The density ρ in $slugs/ft^3$ is given by the equation of state [4].

$$\rho = \left(\frac{p_{SI}}{R_{air}.T_K}\right) \times 0.0019 \quad (1)$$

2.2.2 Curve fitting

Dataplot provided by Lockheed Martin Corporation is used to fit curve to model the relationship between the engine specific fuel consumption (c_{bhp}) and engine horsepower (bhp).

$$c_{bhp} = 4.372 \times 10^{-12} bhp^2 - 2.908 \times 10^{-8} bhp + 1.819 \times 10^{-4} \quad (2)$$

2.3 Performance Variables Calculations

1. **Range**
   Range of the UAV is calculated in accordance with the Breguet Range equation [1] and is given in $ft.$ as

$$R = \frac{550\eta_p}{c_{bhp}} \frac{L}{D} \ln\left[\frac{0.975W_0}{W_0 - W_f}\right] \tag{3}$$

   The 0.975 term is an approximate allowance for additional fuel used during takeoff, climb, descend, and landing. Here the Gross Take Off Weight (GTOW) (W_0) satisfies

$$W_0 = W_{engine} + W_{structure} + W_f + W_{payload} \tag{4}$$

   where W_f is the fuel weight in lbs. The sum of the engine weight W_{engine} and the structure weight $W_{structure}$, also known as the empty weight W_e, is modeled as a fraction of GTOW. The fraction is given [1] by:

$$W_e = aW_0^{0.91} \tag{5}$$

   where a is a coefficent that depends on the type of design. We use $a = 1.15$. Thus,

$$W_0 = aW_0^{0.91} + W_f + W_{payload} \tag{6}$$

   to The $\frac{L}{D}$ ratio in the range equation is given [1] by:

$$\frac{L}{D} = \frac{1}{\frac{qC_{D0}}{W_0/S} + (W_0/S)\frac{K}{q}} \tag{7}$$

   where,

$$C_{D0} = C_{fe}\frac{S_{wet}}{S_{ref}} \tag{8}$$

   and

$$K = \frac{0.424 \times S_{ref}}{b^2} \tag{9}$$

   Another constraint is obtained by looking at the ratio $\frac{S_{wet}}{S_{ref}}$. The ratio is bounded below by 2 due to the definitions of these terms and in practice the ratio is observed [1] to be bounded above by 8. Thus,

$$2 \leq \frac{S_{wet}}{S_{ref}} \leq 8 \tag{10}$$

2. **Stall Speed**
   Stall speed is calculated by equating Lift with the GTOW. Thus,

$$\begin{aligned} W_0 &= L \\ &= qSC_L \end{aligned} \tag{11}$$

   where,

$$q = \frac{1}{2}\rho V_{stall}^2 \tag{12}$$

   Rearranging we have,

$$V_{stall} = \frac{1}{1.689}\sqrt{\frac{2W_0}{S_{ref}\rho C_{L_{max}}}} \tag{13}$$

   The factor 1.689 converts the speed in ft/s to kts.

3. **Endurance**
   Endurance is the length of time spent in cruising flight. For the mission profile under consideration

$$E \approx \frac{R}{V_{cruise}} \tag{14}$$

   At the cruising speed the force balance gives us

$$\begin{aligned} T &= D \\ \Rightarrow \frac{0.75 \times 550 bhp \eta_p}{V_{cruise}} &= q S_{ref}(C_{D0} + K C_L^2) \end{aligned} \tag{15}$$

   where,

$$C_L = C_{L-cruise} = \frac{0.90 \times W_0/S}{q} \tag{16}$$

   Here we assume that the engine's preferred cruise power setting is 75% and at start of the cruise the wing loading is 90% of the take-off wing loading.

4. **Runway Length**
   Raymer [1] provides a graph depicting the variation of Runway Length R_L relative to the Take-off Parameter $T.O.P.$ The following linear fit best describes the relationship between the two.

$$R_L = 7.93 T.O.P - 153.57 \tag{17}$$

   Here,

$$T.O.P. = \frac{1.21 \frac{W_0}{S_{ref}} \frac{W_0}{bhp}}{C_{L_{max}}} \tag{18}$$

5. **Max. Speed**
   Raymer [1] provides an expression for the power loading as a function of the maximum aircraft speed. The present design warrants the use of the following equation

$$V_{max}^{-0.79} \times 680 = \frac{W_0}{0.6 bhp} \tag{19}$$

   where,

$$V_{max} \geq 100 \tag{20}$$

2.4 Model Reduction

In previous section performance variables were expressed in terms of aerodynamic quantities which in turn were functions of the free variables. In this section the performance variables are expressed in terms of the free variables, constants and mission parameters.

Combining equations leading upto equation (1) the density ρ can be expressed ($slugs/ft^3$) as:

$$\rho = \frac{(1 - 6.865 \times 10^{-6} \times h)^{4.256}}{2.897 \times R_{air}} \times 0.0019 \tag{21}$$

1. **Range**
   Combining equations (2), (3), (7), (8) and (9) we have

$$\begin{aligned} R = & \frac{550\eta_p}{4.372 \times 10^{-12} bhp^2 - 2.908 \times 10^{-8} bhp + 1.819 \times 10^{-4}} \\ & \times \left[\frac{1.426 \rho V_{cruise}^2 C_{fe} \left(\frac{S_{wet}}{S_{ref}}\right)}{\left(\frac{W_0}{S_{ref}}\right)} + \frac{0.424 W_0}{2.853 \rho b^2 V_{cruise}^2} \right]^{-1} \ln \left[\frac{0.975 W_0}{W_0 - W_f} \right] \end{aligned} \tag{22}$$

where,

$$W_0 = aW_0^{0.91} + W_f + W_{payload} \tag{23}$$

and, V_{cruise} is expressed in kts.

2. **Stall Speed**

$$V_{stall} = \frac{1}{1.689}\sqrt{\frac{2W_0}{S_{ref}\rho C_{L_{max}}}} \tag{24}$$

3. **Endurance**
Combining equations (14) and (22) we have the expression for endurance in hours as:

$$\begin{aligned} E &= \frac{R}{V_{cruise}} \\ &= \frac{550\eta_p}{4.372\times 10^{-12}bhp^2 - 2.908\times 10^{-8}bhp + 1.819\times 10^{-4}}\left[\frac{1.426\rho V_{cruise}^2 C_{fe}\left(\frac{S_{wet}}{S_{ref}}\right)}{\left(\frac{W_0}{S_{ref}}\right)} + \frac{0.424W_0}{2.853\rho b^2 V_{cruise}^2}\right]^{-1} \\ &\quad .\ln\left[\frac{0.975W_0}{W_0 - W_f}\right]\frac{1}{1.689V_{cruise}}\frac{1}{3600} \end{aligned} \tag{25}$$

Combining equations (15) and (16) and rearranging

$$4.069\rho C_{fe}S_{wet}V_{cruise}^4 - 928.95bhp\eta_p V_{cruise} + \frac{0.687W_0^2}{\rho b^2} = 0 \tag{26}$$

Here V_{cruise} is expressed in kts.

4. **Runway Length**

$$R_L = 7.93T.O.P - 153.57 \tag{27}$$

where,

$$T.O.P. = \frac{1.21\frac{W_0}{S_{ref}}\frac{W_0}{bhp}}{C_{L_{max}}} \tag{28}$$

5. **Max. Speed**

$$V_{max}^{-0.79}\times 680 = \frac{W_0}{0.6bhp} \tag{29}$$

where,

$$V_{max} \geq 100 \tag{30}$$

3 Performance Space Characterization using Matlab

With the proposed model Matlab can be used for computation of the performance variables. Specifically, the following charts were generated.

1. S_{wet} and bhp level curves in $Endurance - V_{cruise}$ space.
The following values of free parameters are used for generating the level curves. The bhp level curves are obtained by varying S_{wet} over the given range and vice-versa for the S_{wet} level curves.

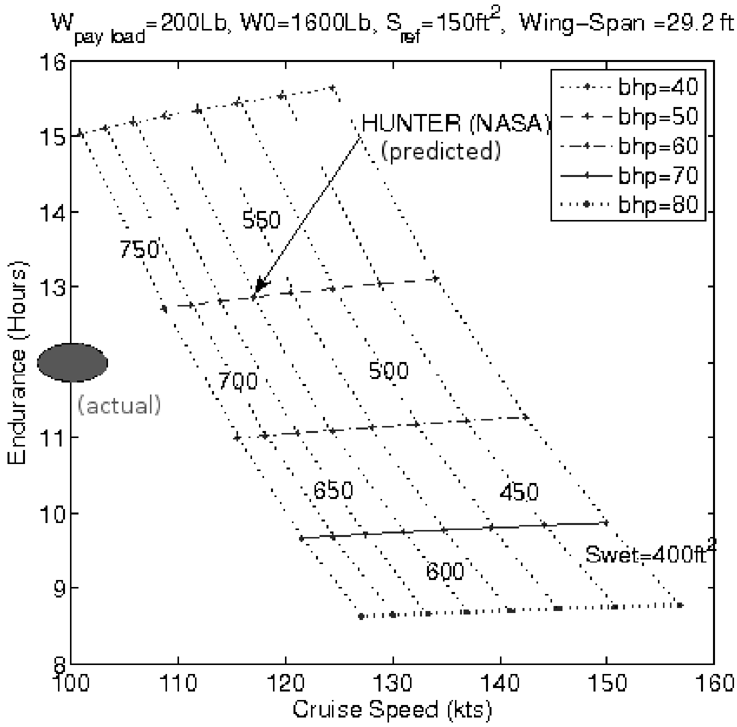


Figure 1: S_{wet} and bhp level curves in $Endurance - V_{cruise}$ space

Free Parameter	Value/Range
$W_{payload}$	200 $lb.$
W_0	1600 $lb.$
S_{ref}	150 ft^2
b	29.2 $ft.$
S_{wet}	[400,700] ft^2
bhp	[40,80] hp

2. W_0 and bhp level curves in $Endurance - V_{stall}$ space.
The following values of free parameters are used for generating the level curves. The bhp level curves are obtained by varying W_0 over the given range and vice-versa for the bhp level curves.

Free Parameter	Value/Range
$W_{payload}$	200 $lb.$
S_{wet}	600 ft^2
S_{ref}	150 ft^2
b	29.2 $ft.$
W_0	[1000,1800] ft^2
bhp	[40,80] hp

3. b and h level curves in $Endurance - V_{cruise}$ space
The following values of free parameters are used for generating the level curves. The b level curves are obtained by varying h over the given range and vice-versa for the h level curves.

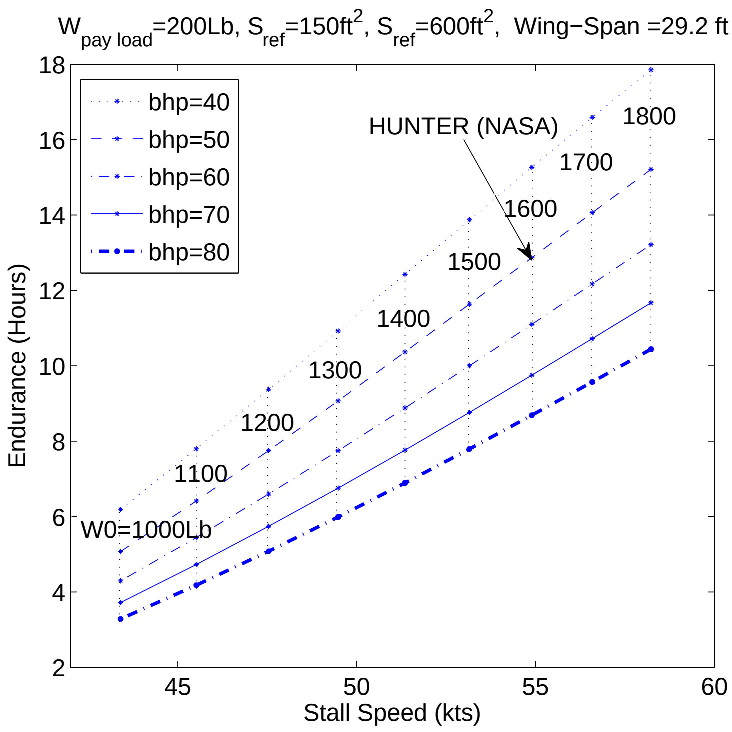


Figure 2: W_0 and bhp level curves in $Endurance - V_{stall}$ space

Free Parameter	Value/Range
$W_{payload}$	200 $lb.$
S_{wet}	600 ft^2
S_{ref}	150 ft^2
W_0	1600 $lb.$
bhp	50 hp
b	[10,35] ft
h	[8000,20000] ft

4. bhp and S_wet level curves in $Endurance - V_{cruise}$ space
   The following values of free parameters are used for generating the level curves. The bhp level curves are obtained by varying S_wet over the given range and vice-versa for the h level curves.

Free Parameter	Value/Range
$W_{payload}$	200 $lb.$
b	10 ft
S_{ref}	150 ft^2
W_0	1600 $lb.$
bhp	[40,80] ft
S_{wet}	[400,750] ft

In the charts, the performance of NASA's Hunter UAV is predicted by the use of our model. However, the prediction on the cruise speed is out of the allowable error margin. This indicates a problem with the model. However, the problem could not be resolved within the given amount of time.

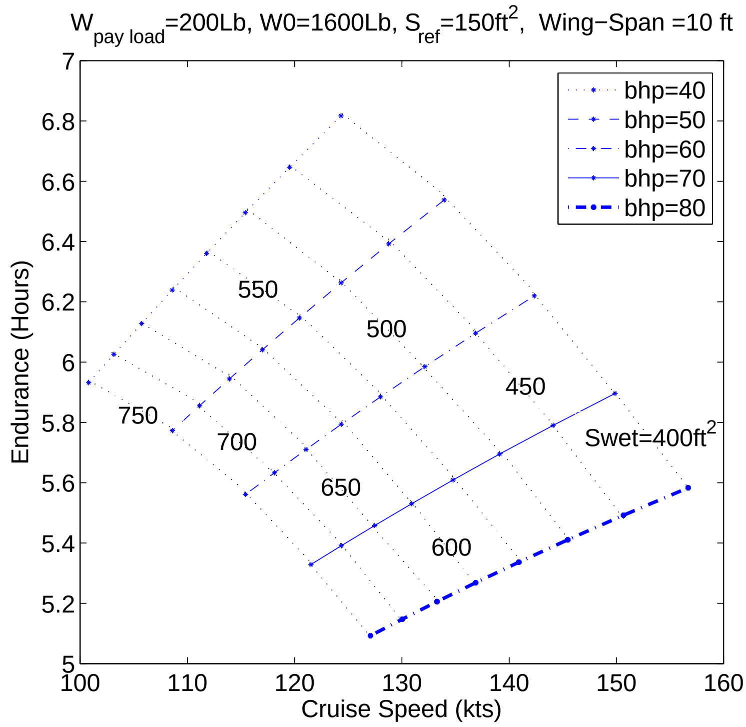


Figure 3: b and h level curves in $Endurance - V_{cruise}$ space

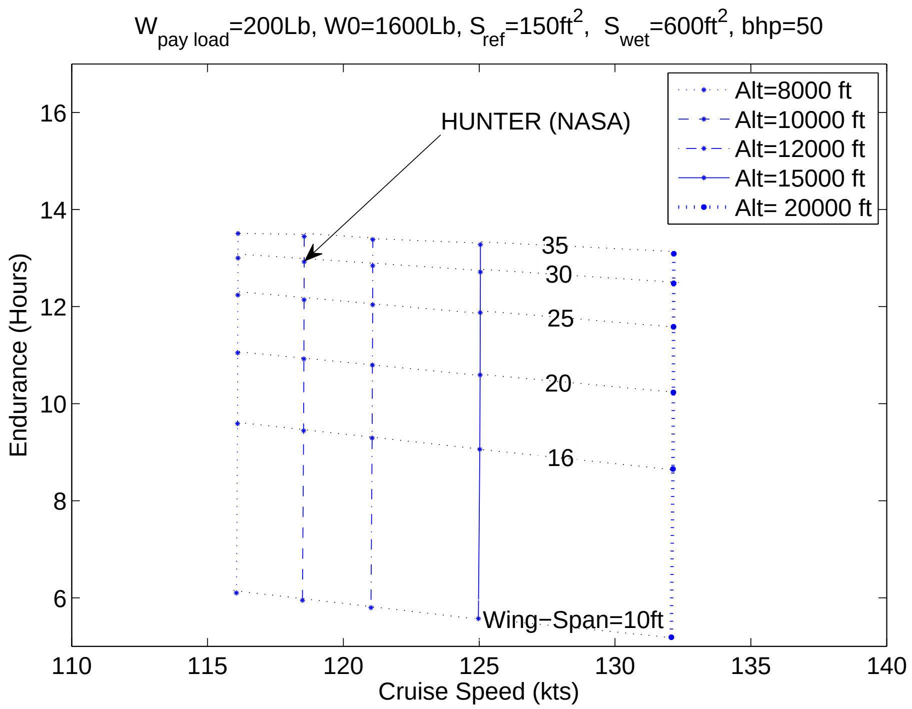


Figure 4: bhp and S_wet level curves in $Endurance - V_{cruise}$ space

4 Performance Space Characterization using BONMIN

BONMIN stands for Basic Open-source Nonlinear Mixed INteger programming. It was chosen over other MINLP solvers for three reasons. Developed by a team of researchers at IBM and Carnegie Mellon University BONMIN is one of the mainstream MINLP solvers. Secondly, the program contains many options which let the users improve performance based on individual problems. These options include four different algorithms that can be implemented to solve MINLP problems. Lastly, BONMIN was used because of its ease. Together with AMPL (A Modeling Language for Mathematical Programming), a nice interface for BONMIN, command files can be written easily and executed quickly.

The performance variable functions in the model and the domain of the free parameters is nonconvex. BONMIN recommends the use of the branch and bound algorithm to solve such problems. Following section presents a discussion of the Mixed Integer Non-linear Programming Problems and the algorithms available for solving those.

4.1 Mixed Integer Non-linear Programming

Mixed Integer Nonlinear Programming(MINLP) problems are conveniently expressed as

$$\begin{array}{rl} \min & z \\ s.t. & f(x,y) \le z \\ & c(x,y) \le 0 \\ x \in X \subseteq \mathbb{N}^n, & y \in Y \subseteq \mathbb{R}^p, \quad z \in \mathbb{R} \end{array} \tag{31}$$

where f, g are twice continuously differentiable functions, and x and y are continuous and discrete variables, respectively. The discrete part x can be restricted to $0, 1$ values, $x \in \{0,1\}^n$. We consider the branch and bound method (BB) and Outer-Approximation (OA) method to solve MINLP problems.

4.1.1 Branch and Bound

The Branch and Bound(BB) method starts by solving first the continuous NLP relaxation. If all discrete variables take integer values the search is stopped. Otherwise, a tree search is performed in the space of the integer variables. These are successively fixed at the corresponding nodes of the tree, giving rise to relaxed NLP subproblems which yield lower bounds for the subproblems in the descendant nodes. Fathoming of nodes occurs when the lower bound exceeds the current upper bound, when the subproblem is infeasible or when all integer variables x_i take on discrete values. The latter yields an upper bound to the original problem.

The BB method is generally only attractive if the NLP subproblems are relatively inexpensive to solve, or when only few of them need to be solved. This could be either because of the low dimensionality of the discrete variables, or because the integrality gap of the continuous NLP relaxation of (P1) is small.

This tree example illustrates the BB method. Consider all x as continuous variables (relaxation) and solve the NLP problem, we obtain as solution

$$\begin{array}{rcl} z & = & 5.8 \\ (x_1, x_2, x_3) & = & (0.2, 1, 0) \end{array}$$

Note that 5.8 is a lower bound on the optimal solution of the MINLP. In the first level choose a non-integer variable of the optimal solution of this problem(x_1 in following picture) and fixed as 0 or 1 and solve subproblems 2 and 3. Since the optimal value of 2 is smaller than the optimal value of 3, fixed x_3 as 0 or 1 and solve them(subproblems 4 and 5). Since a subproblem 4 is infeasible and the optimal value of subproblem 5 is smaller than the optimal value of 3, backtrack the subproblem 3 and solve subproblems 6 and 7 with fixed variable x_2. Using this process, we can find a optimal value $z = 8$.

The BB method is attractive if the NLP subproblems can be solved. If the low dimensionality of the discrete variables, or the continuous NLP subproblem of 31 is small, this could be good method.

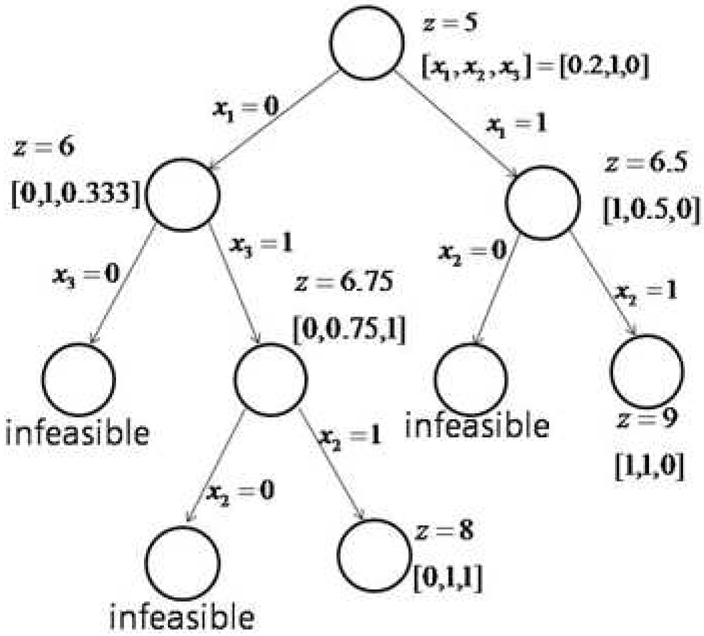


Figure 5: Tree

4.1.2 Outer-Approximation

Outer Approximation(**OA**) algorithm is worked with the condition:
C1 : **X** and **Y** are a nonempty, compact and convex set. And the functions

$$f : \mathbb{R}^n \to \mathbb{R}$$
$$g : \mathbb{R}^n \to \mathbb{R}^p$$

are convex in **Y**.
C2 : Both f and g are differentiable.
C3 : A constraint qualification hold at the solution of every nonlinear programming problem resulting from (31) by fixing y.

Note that the assumption includes (i) Separability in $\mathbf{x}$ and $\mathbf{y}$ and (ii) Linearity in $\mathbf{y}$.

The OA algorithm consists of solving an alternative sequence of Nonlinear Programming (NLP) optimization subproblems and Mixed-Integer Linear Programming (MILP) master problems. The search of a convex problem is terminated when the predicted lower bound exceeds the upper bound, otherwise it is terminated when the NLP solution can not be improved.

We will consider the following NLP subproblem(saying primal problem)

$$\begin{array}{ll} \min & z = f(x_k, y) \\ s.t. & g(x_k, y) \leq 0 \\ & y \in Y \end{array} \tag{32}$$

for some fixed x_k. Note that the solution of this problem is a upper bound of original problem(31).

The OA algorithm is based on the following theorem (Duran and Grossmann, 1986):

Theorem 1 *Problem 31 and the following MILP master problem have the same optimal solution* (x^*, y^*),

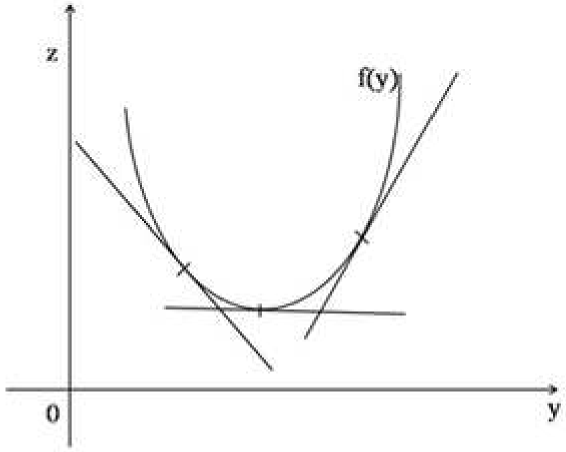


Figure 6: Linear Approximation

$$\begin{array}{ll} \min & z = \alpha \\ & \alpha \geq f(x_k, y_k) + \nabla f(x_k, y_k) \begin{bmatrix} x - x_k \\ y - y_k \end{bmatrix} \\ s.t. & \\ & 0 \geq g(x_k, y_k) + \nabla g(x_k, y_k) \begin{bmatrix} x - x_k \\ y - y_k \end{bmatrix} \\ & k \in K^* \end{array} \tag{33}$$

where $K^* = \{k|k$ for all feasible $y_k \in Y, (x_k, y_k)$is the optimal solution to the primal problem$\}$.

Since the master problem requires the solution of all feasible discrete variables x_k, the following MILP relaxation is considered, assuming that the solution of K NLP subproblems is available:

$$\begin{array}{ll} \min & z = \alpha \\ & \alpha \geq f(x_k, y_k) + \nabla f(x_k, y_k) \begin{bmatrix} x - x_k \\ y - y_k \end{bmatrix} \\ s.t. & \\ & 0 \geq g(x_k, y_k) + \nabla g(x_k, y_k) \begin{bmatrix} x - x_k \\ y - y_k \end{bmatrix} \\ & k = 1, 2, \cdots K \end{array} \tag{34}$$

Given the assumption on convexity of the functions $f(x, y)$ and $g(x, y)$, we can know following property. The solution of master problem(34), corresponds to a lower bound of the solution of problem(31). The master problem of the OA can be interpreted geometrically by examinig the effect of the linear support function on the objective function and the constraints. Following figure shows the linear supports of the objective function taken at three points. The natural approach of solving the master problem is relaxation; that is , consider at each iteration the linear supports of the objective and constraints around all previously linearization points. This way, at each iteration a new set of linear support constrains are added which improve the relaxation and therefore the lower bound. The algorithm of OA are as following since function linearizations are accumulated as iterations proceed, the master problems (34) yield a non-decreasing sequence of lower bounds.

The OA algorithm consists of performing a cycle of major iterations, $k = 1, ..., K$, in which (32) is solved for the corresponding x_k and the relaxed MILP master problem (34) is updated and solved with the corresponding function linearizations at the point (x_k, y_k).

Duran and Grossmann proved that the finite convergence of OA algorithm(An outer approximation algorithm for a class of mixed-integer nonlinear programs. Math.Prog. 1986a) under the conditions C1, C2 and C3. So the OA method does not work if one of the conditions is not satisfied.

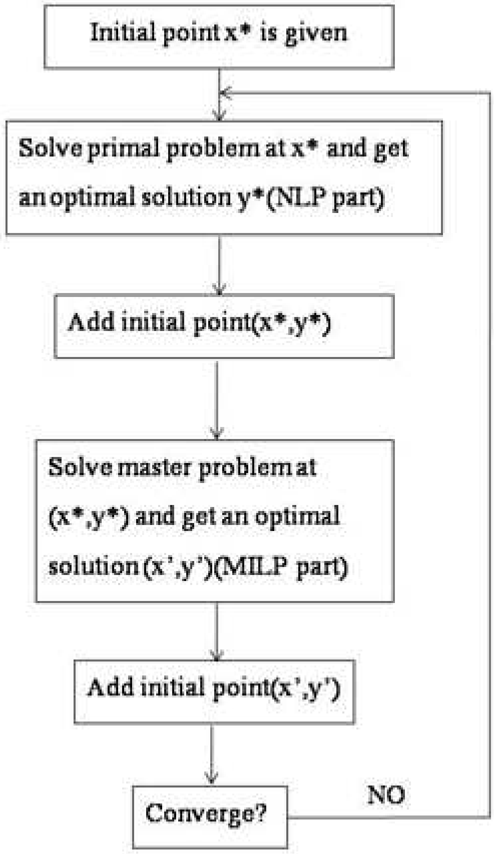


Figure 7: Algorithm

4.2 Use of BONMIN

Before using BONMIN for our model we checked the validity of BONMIN solutions. The following simple example was solved by hand and then run in BONMIN. The results were identical.

$$\begin{aligned} \text{minimize } f(x,y) &= -x - y_1 - y_2 \\ &\text{subject to} \\ (y_1 - 1/2)^2 + (y_2 - 1/2)^2 &\leq 1/4 \\ x - y_1 &\leq 0 \\ x + y_2 + z &\leq 2 \\ y &\in \mathbb{R}^2 \\ x &\in \{0,1\} \\ z &\in \{0,1,2,3,4,5\} \end{aligned}$$

This problem has solution $x^* = 1$, $z^* = 0$, and $y^* = [1, 1/2]^T$ with minimum function value $f(x^*, y^*) = -2.5$.

As seen in the previous section the design space can be explored by analyzing plots of endurance vs cruising velocity, range vs velocity, and various contour plots. With BONMIN we can allow all but one of the variables to vary simultaneously and explore the optimal points in the design space. The procedure can be explained in the light of Figure 8 (Appendix) as follows. First the Endurance Equation 25 is simultaneously optimized for a particular value of V_{cruise}. The latter is gradually increased and optimal endurance values corresponding to each value of V_{cruise} are plotted. This gives us the optimal endurance curve against V_{cruise}. Next, this $Endurance - V_{cruise}$ space can be explored by plotting contours of fuel weight. For this, a particular fuel weight is chosen and then optimal endurances are obtained relative to every value of V_{cruise}. Figures 9 through 11 (in Appendix) are obtained in a similar fashion.

4.2.1 Critique of BONMIN

BONMIN has a few drawbacks. Installing BONMIN and AMPL is not an easy task (required most of an afternoon). When BONMIN is run through online servers, the time it takes to receive results is a function of server load. This usually ranges from a few seconds to many hours.

5 FLOPS

FLOPS (FLight OPtimization System) is a software package that was developed at NASA for conceptual design and evaluation of aircraft. It allows a design space of up to 18 parameters and uses a combination of physical equations and empirical data fits to determine the best type of plane to perform a given mission. FLOPS uses a Penalty Method along with a BFGS Quasi-Newton method to solve the optimization problem.

Outline of the FLOPS model:

FLOPS can optimize the design by changing:

- Gross Weight
- Engine Thrust
- Wing Taper Ratio
- Wing Sweep Angle
- Average Wing Thickness to Chord Ratio
- Reference Wing area
- Wing Aspect Ratio
- Cruise Altitude
- Cruise Mach Number

- Other variables related to specific engine design and noise generation

To evaluate the objective function for each point in the design space, FLOPS does the following:

1. Estimates the weight and exposed surface areas of all the components using the geometry of the aircraft. A structural analysis model is used to determine wing weight, as it is easily the most complicated part of the aircraft.
2. Calculates the maximum volume and weight of fuel the current design can carry.
3. Generates a table of various thrust and fuel consumption values for the type of engine specified, using data from the engine design module.
4. Creates tables of lift-induced drag and parasitic drag coefficient for various mach numbers, lift coefficients, and altitudes. This uses information from the airplane geometry, [6], and [5].
5. Generates a range of operational weights, and then determines the dynamic pressure and lift coefficient for each weight at our cruising speed and altitude.
6. Based on these, it determines the drag and lift forces on the plane for each weight.
7. FLOPS then determines the thrust and fuel consumption for each weight by interpolating the engine tables generated above.
8. Using this information, FLOPS now knows the aircraft state at any Mach number, weight, or altitude expected on its mission. It then integrates the equations of motion along its mission path to calculate performance results.
9. The objective function takes the form

$$\begin{aligned} f(\vec{x_d}) =& w_f(\text{total fuel}) + w_r(\text{Range}) + w_M(V_{cruise})(\frac{L}{D}) \\ &+ w_{gw}(\text{Gross Weight}) + w_{fc}(\text{Fuel Consumption}) \end{aligned}$$

   Where the w are weights specified by the user.

FLOPS then employs a Penalty Method to this objective function. A penalty method transforms the constrained problem

$$\min_x f(x)$$
$$c_i(x) \leq 0$$

to the unconstrained problem

$$\min_x f(x) + \mu \sum_i P(c_i(x))$$

Where P is the *Penalty Function.* FLOPS uses the Fiacco-McCormick Penalty function, which looks like:

$$P(x_i) = \frac{1}{1 - \frac{x_i}{U}} \qquad \text{For the upper bound of } x_i$$
$$P(x_i) = \frac{1}{1 - \frac{L}{x_i}} \qquad \text{For the lower bound of } x_i$$

FLOPS then solves this unconstrained problem using BFGS, which is a superlinearly convergent quasi-Newton method. After the BFGS method converges, μ is successively decreased and the resulting problems are again solved by BFGS, resulting in a final (numerically) optimal solution. (For more info on BFGS and other Quasi-Newton approaches, see [7].)

5.1 Critique of FLOPS

Flops uses very sophisticated models for many of its components, and the empirical data it uses can be replaced due to changes in technology/experiments/etc. It allows for a wide range of designs - anything from passenger planes to combat fighters.

On the other hand, FLOPS requires you to know a large amount of information about your plane to begin with, and is mainly concerned with optimizing the wing shape, engine output, and amount of fuel. There are many other basic design characteristics that we may want to look for, such as number and location of engines, type of engine, or presence and type of tails.

FLOPS does not contain a model for aircraft stability and maneuverability, and does a poor job at incorporating the effects of the tails. It also does a poorer job for both very low speed and very high speed aerodynamics. Computation-wise, the biggest bottleneck in FLOPS is its noise footprint calculations, which we are not concerned with ([11]).

It may be possible to take the ideas of the very general FLOPS model and repurpose them for our specific case (or other roughly similar subsonic missions). Such a tailoring may be able to reduce the amount of computational time spent within FLOPS, while still maintaining its excellent flight model..

6 Conclusions and future directions

We developed a low-fidelity aerodynamic model for UAV design. However the model predicted values are not within our 85% accuracy goal and as such work needs to be done for improving the model. The model also needs to incorporate volume constraints and some of the discrete variables like number of engines. We developed a near real time technique based on BONMIN for visualisation of the design space. The process can be made real time by using automated scripts that efficiently incorporate BONMIN into packages such as Matlab.

7 Acknowledgements

We would like to thank the IMA for creating this wonderful interaction avenue among industry experts and students. We appreciate the hospitality and the stimulating work environment. We would also like to thank our mentor Dr. John Hoffman for presenting a very interesting and challenging problem and motivating us to do our best.

References

[1] D. Raymer (2007). *Simplified Aircraft Design for Homebuilders,* Design Dimension Press, Los Angeles, CA.

[2] D. Raymer (1999). *Aircraft Design: A Conceptual Approach,* AIAA Education Series, Reston, VA.

[3] Glenn Research Center. *Earth Atmosphere Model* [www http://www.grc.nasa.gov/WWW/K-12/airplane/atmos.html].

[4] Glenn Research Center. *Equation of State (Ideal Gas)* [www http://www.grc.nasa.gov/WWW/K-12/airplane/eqstat.html].

[5] Simon C. Sommer and Barbara J. Short. Free-flight measurements of turbulent-boundary-layer skin friction in the presence of severe aerodynamic heating at Mach numbers from 2.8 to 7.0. NACA Technical Note 3391, 1955.

[6] R.D. Elliot. Empirical Drag Estimation Technique (EDET) User's Manual. Lockheed California, Report LR28788, Burbank, CA, Dec. 1978.

[7] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* Springer, second edition, 2006

[8] *NEOS Server for BONMIN* [http://neos.mcs.anl.gov/neos/solvers/minco:Bonmin/AMPL.html]

[9] *AMPL Modeling Language for Mathematical Programming* [http://www.ampl.com/]

[10] *BONMIN Homepage* [https://projects.coin-or.org/Bonmin]

[11] L. A. (Arnie) McCullers, personal communication

A Glossary of Aerodynamical Terms

Following is a glossary of the aerodynamical terms used in this work.

- Wetted Area
  Wetted Area is defined as the total surface area of the aircraft, including the top and bottom of the wings, the top, sides, and bottom of the fuselage, and both sides of the tails.
- Reference Aerodynamic Area
  The reference area is usually the plan-form or flat projection (the wing's shadow at noon) area of the wing.
- Skin friction coefficent
  Skin friction coefficient takes into account the overall aerodynamic cleanness of an aircraft. Aerodynamic cleannes is defined by the amount of excrescence drag on an aircraft.
- Maximum Lift Coefficient
  The lift generated by an airfoil is proportional to the dynamic pressure of the fluid flow around the airfoil, and the planform area of the airfoil. The constant of proportionality is referred to as the lift coefficent. In an aircraft the lift coefficient at the stall speed is maximum and is called the maximum lift coefficent.

B Performance Characteristics using BONMIN

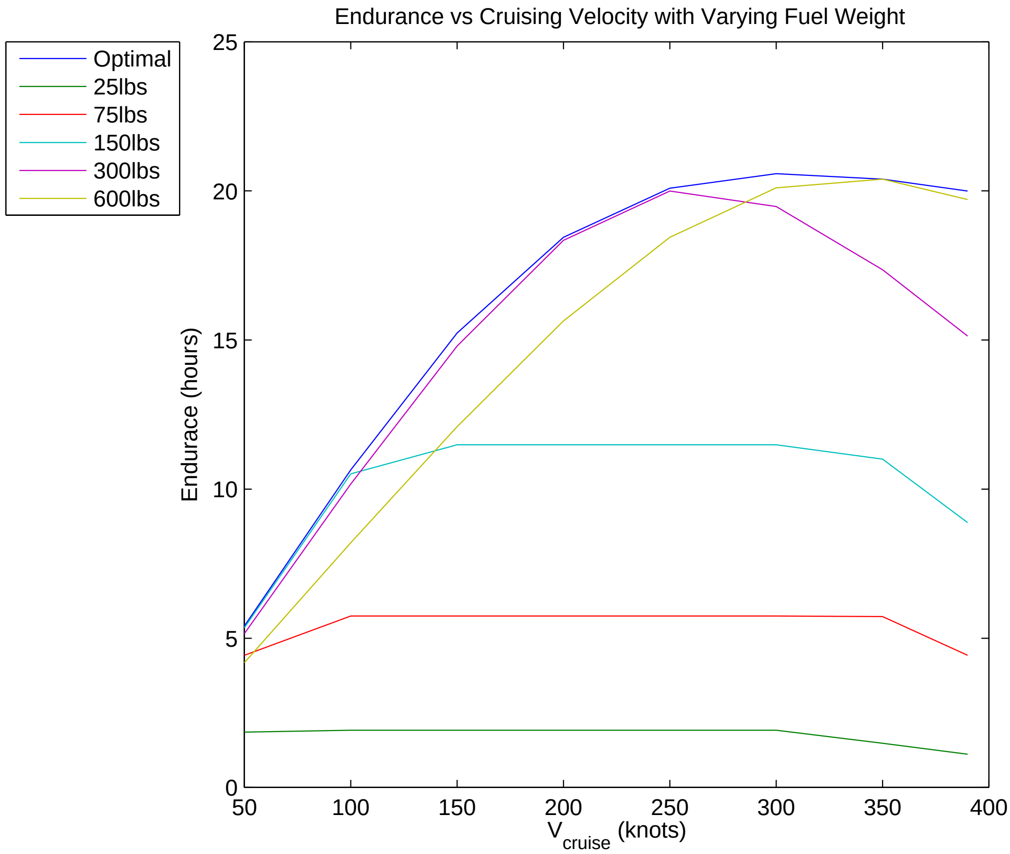


Figure 8: $Endurance - V_{cruise}$ with varying fuel weight

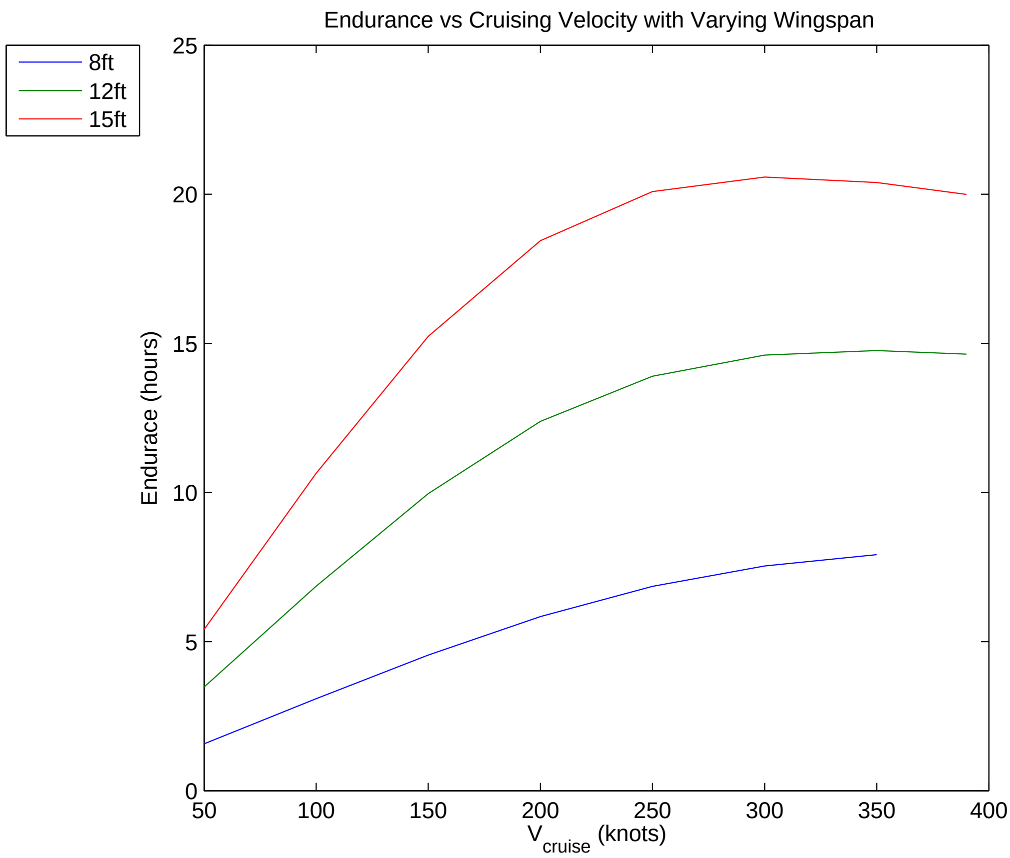


Figure 9: $Endurance - V_{cruise}$ with varying wing span

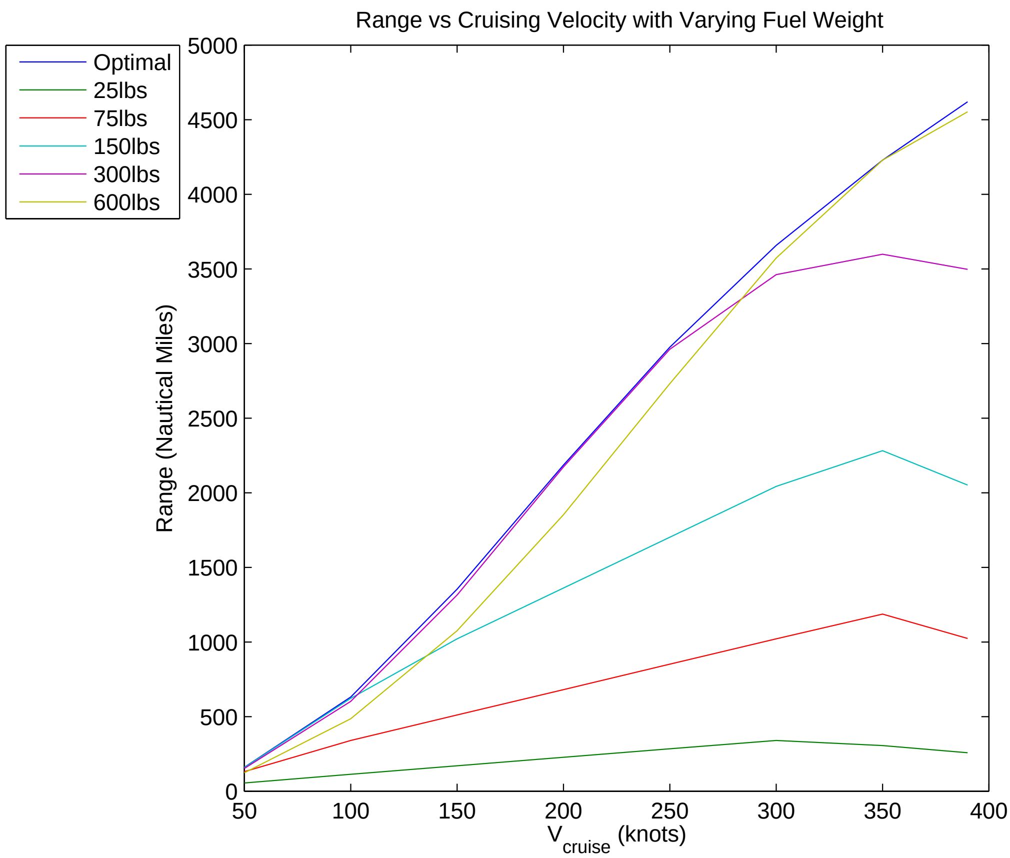


Figure 10: $Range - V_{cruise}$ with varying fuel weight

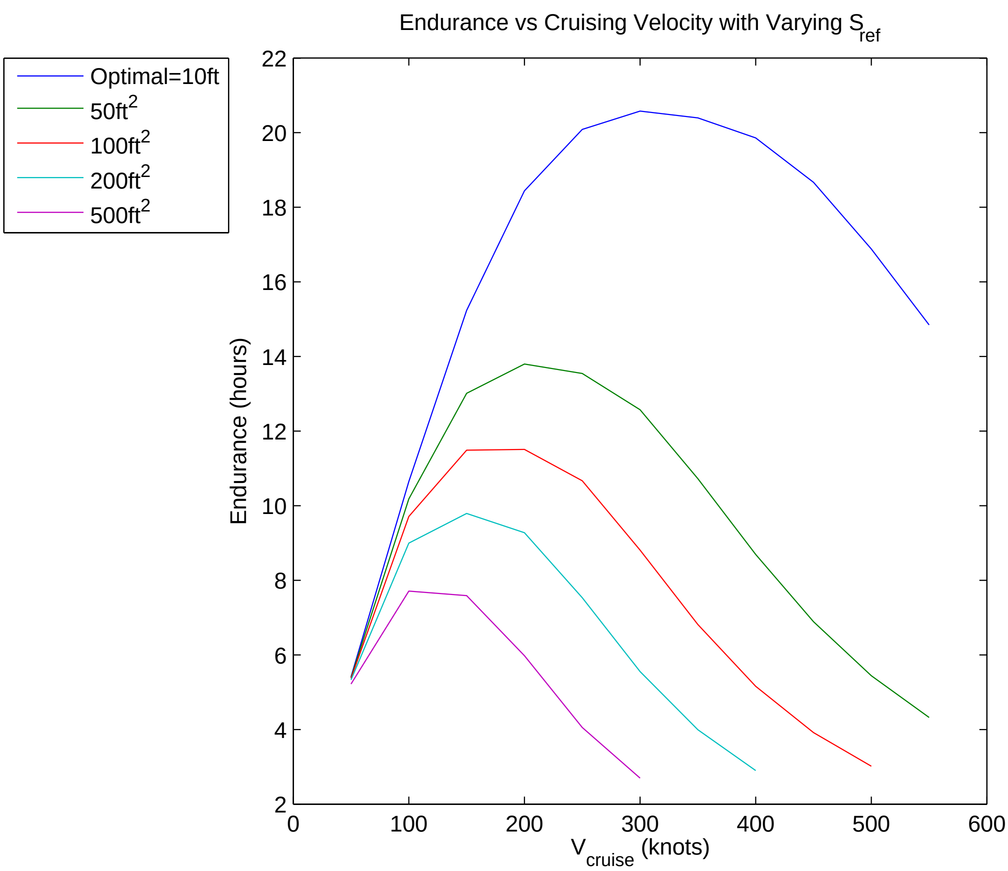


Figure 11: $Endurance - V_{cruise}$ with varying S_{ref}