Contents

1 Aircraft Performance Optimization **2**
1.1 Development of a Gradient-Based Optimization Algorithm 2
1.2 Discipline and Design Space Exploration 5
1.2.1 Aerodynamic Discipline - Efficiency Optimization 5
1.2.2 Weight Estimation Discipline - MTOW Optimization 7
1.2.3 Performance Discipline Optimization: STO, fuel, time 8
1.2.4 Structural Discipline Formulation 12
1.3 Multi-Disciplinary Design Optimization (MDO) 14
1.3.1 Direct Optimization with Composite Objective Function 14
1.3.2 Response Surface Methodology (RSM) 16
1.3.3 Optimization with Surrogate Models 17
1.3.4 Multi-Objective Pareto Analysis 19
1.3.5 Pareto Front Analysis (3 Design Variables) 23
1.3.6 Interactive Visualization Support Tool 26

1. Aircraft Performance Optimization

The objective of this chapter is to optimize the aircraft performance by evaluating the effects of different objective functions, constraints, and design variables. The study progresses from simple mono-disciplinary analyses to more complex Multi-Disciplinary Design Optimization (MDO) problems. Initially, a custom optimization algorithm was developed in Python, based on the methodologies presented in *Engineering Design Optimization* by Martins and Ning. Following the validation of this algorithm, the focus shifted to the aircraft optimization process, utilizing an Excel spreadsheet via the Python *xlwings* library. The Excel spreadsheet compute the aerodynamic and the performance of the aircraft, mostly through semi-empirical formulas. Initial optimizations were performed using both the custom algorithm and the SciPy library to benchmark results and computational performance. Then thanks to surrogates models built on the excel spreadsheet, it has been possibile to obtain the pareto fronts of the objectives chosen and a interactive visualization app of the optimal design was built in python and can be found on GitHub (link).

1.1 Development of a Gradient-Based Optimization Algorithm

To address the design problems effectively, a custom gradient-based optimization routine was implemented. The core structure of the algorithm relies on the modular approach described by Martins and Ning in their book. The implementation avoids the use of "black-box" solvers to ensure full control over the convergence behavior and constraint handling. The algorithm operates through the following macro-blocks:

Gradient Estimation and Search Direction

The algorithm utilizes a numerical differentiation approach. The gradient $\nabla f(x)$ is computed using the **forward difference method**. To determine the search direction p_k at iteration k, the method employs a **Conjugate Gradient** strategy rather than a simple **Steepest Descent**. This choice reduces the "zig-zagging" phenomenon often observed in narrow valleys. The direction is updated as:

$$p_k = -\nabla f(x_k) + \beta_k p_{k-1}$$

where β_k is a scalar calculated to ensure conjugacy between successive directions, improving convergence speed near the optimum.

Line Search and Wolfe Conditions

Once a descent direction p_k is identified, the algorithm determines the step size α_k to update the current point $x_{k+1} = x_k + \alpha_k p_k$. An inexact line search is implemented to satisfy the **Strong**

Wolfe Conditions, which guarantee both a sufficient decrease in the objective function+ and a sufficient curvature reduction. The step size α is accepted if:

$$f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T p_k \quad \text{(Sufficient Decrease)}$$

$$\left|\nabla f(x_k + \alpha p_k)^T p_k\right| \leq c_2 \left|\nabla f_k^T p_k\right| \quad \text{(Curvature Condition)}$$

where $0 < c_1 < c_2 < 1$. The algorithm employs a *pinpointing* strategy (utilizing cubic interpolation) to efficiently locate an α that satisfies these criteria without evaluating the function excessively.

Constraint Handling: The Penalty Method

The algorithm handles constraints using an **Exterior Penalty Method**. This transforms the constrained optimization problem into a sequence of unconstrained problems. A composite objective function $\Phi(x)$ is defined by adding penalty terms proportional to the square of the constraint violations:

$$\Phi(x, \rho) = f(x) + \sum_i \rho_{eq,i} [h_i(x)]^2 + \sum_j \rho_{ineq,j} [\max(0, g_j(x))]^2$$

where $h_i(x) = 0$ represents equality constraints and $g_j(x) \leq 0$ represents inequality constraints. The penalty coefficients ρ are increased iteratively in an outer loop if the constraints are not satisfied within a specified tolerance, forcing the solver to converge towards the feasible region.

Algorithm Validation and Globalization Strategy

Before applying the algorithm to the aircraft design, a validation phase was conducted using the Ackley function. This function is characterized by a nearly flat outer region and a large hole at the center, modulated by many small local minima, making it ideal benchmark for testing global optimization capabilities. The mathematical formulation used is:

$$f(x, y) = -20 \exp\left(-0.2\sqrt{0.5(x^2 + y^2)}\right) - \exp\left(0.5(\cos(2\pi x) + \cos(2\pi y))\right) + e + 20$$

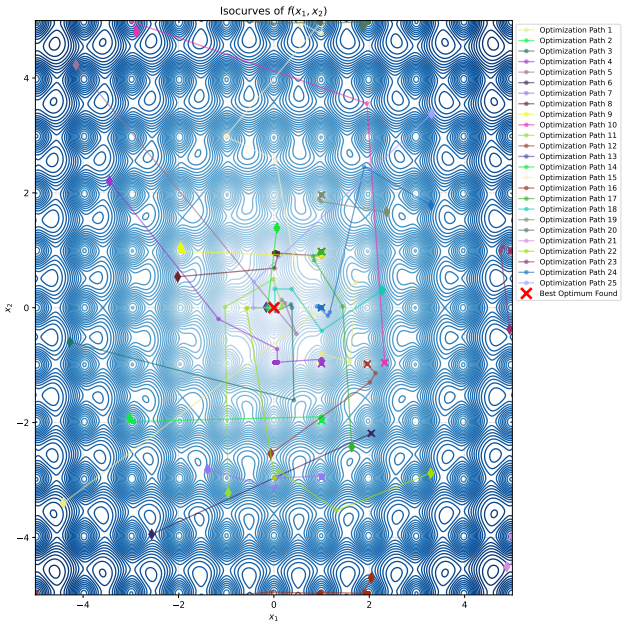
A standard gradient-based algorithm is prone to getting trapped in local minima. To overcome this limitation, a **globalization strategy** was implemented based on a **Multi-Start approach**.

The design space was discretized into a grid of 25 rectangular sub-regions (5×5). For each sub-region, a random starting point was generated, and a local optimization was performed. The final solution was selected as the minimum among all converged optima. This method allows a local optimizer to explore the design space globally. The computational cost of evaluation such analytical function is negligible, so no other comparison between the algorithms has been made in terms of performance.

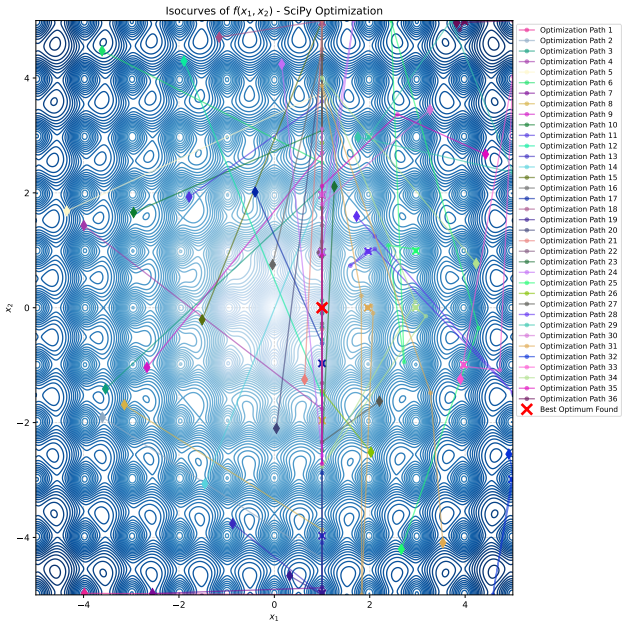
The results of this validation are presented in Figure 1.1, computed with both the Custom and SciPy algorithm.

For the **unconstrained case** the custom algorithm (Fig. 1.1a) successfully identified the global minimum at $(0, 0)$, demonstrating the effectiveness of the multi-start approach, while the SciPy implementation (Fig. 1.1b) identified the best optimum at a local minimum point right next to the global minimum. This does not mean that the custom algorithm is better, but it highlights the importance of knowing the algorithm used in order to make the most of it and achieve better results.

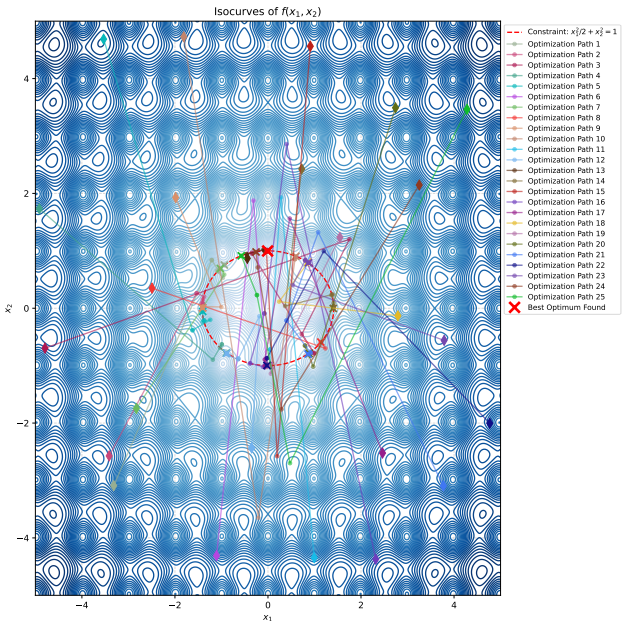
For the **constrained cases** the penalty method was tested with an equality constraint on an ellipse (Fig. 1.1c) and an inequality constraint requiring the solution to lie outside the ellipse (Fig. 1.1d). In both cases, the algorithm .



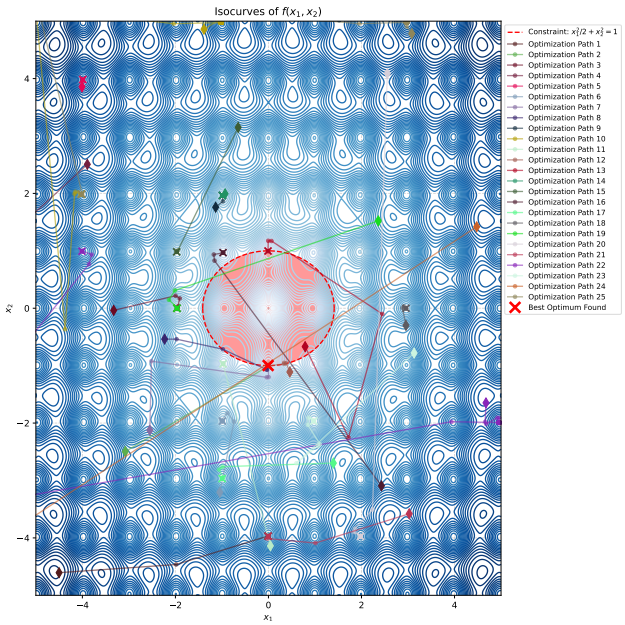
(a) Unconstrained optimization (Custom)



(b) Unconstrained optimization (SciPy)



(c) Equality Constraint: $x_1^2/2 + x_2^2 = 1$



(d) Inequality Constraint: Outside Ellipse

Figure 1.1: Validation of the globalized optimization strategy on the Ackley function. The lines represent the optimization paths starting from different sub-regions of the domain.

1.2 Discipline and Design Space Exploration

The study progressed to the application on the specific aircraft design problem. To ensure a robust understanding of the design space and the solver's behavior, the complexity of the optimization problem was increased incrementally. The process began with single-discipline optimizations, varying objective functions, design variables, and constraints, before advancing to the full Multi-Disciplinary Design Optimization (MDO).

A crucial aspect of the next problems formulations is **the normalization** of inputs, outputs, and constraints. If design variables or constraint values differ by several orders of magnitude, the optimization becomes ill-conditioned, causing the solver to converge slowly or fail. In this studies, both the design variables and the objective functions were scaled to be of order 1. Similarly, the constraints were normalized by their reference values to ensuring that constraint violations are comparable in magnitude to the objective function gradients. The optimizations utilizes the Excel module developed, interfaced via Python. The geometry of the wing has been varied while maintaining constant relative ratios such as the ratio of the tip chord to the root chord.

1.2.1 Aerodynamic Discipline - Efficiency Optimization

The first disciplinary case study focuses on aerodynamics. The primary goal is to maximize the aircraft's **max trimmed aerodynamic efficiency**($E = L/D$). To formulate the problem according to the standard convention (minimization), the objective function is defined as the negative of the efficiency ratio. Two design variables were selected: the wing span ratio and the root chord ratio. These act as multipliers of the baseline aircraft geometry. The optimization problem can be formally stated as follows:

$$
\begin{aligned}
\underset{x}{\text{minimize}} \quad & f(x) = -\frac{E_{trimmed}(x)}{E_{ref}} \\
\text{subject to} \quad & g_1(x) = \frac{0.25 - SSM(x)}{SSM_{ref}} \leq 0 \\
& g_2(x) = \nu_h(x) - 1.1 \leq 0 \\
& g_3(x) = 0.8 - \nu_h(x) \leq 0 \\
& x_{lower} \leq x \leq x_{upper}
\end{aligned}
$$

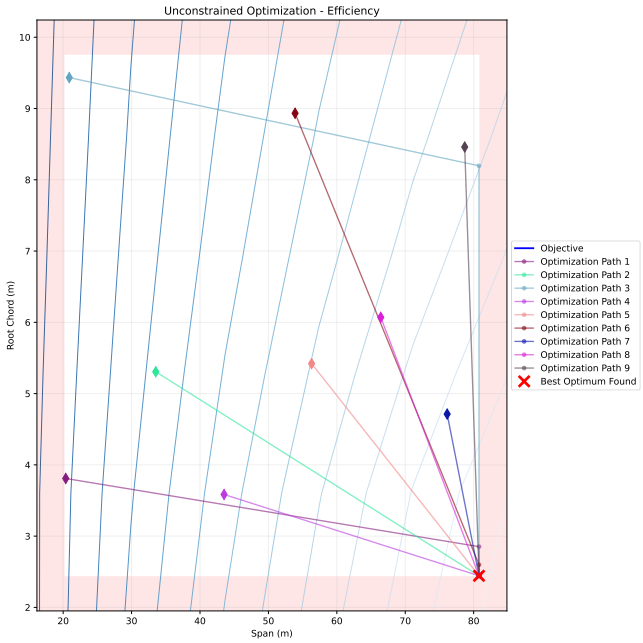
where $x = [x_1, x_2]^T$, with x_1 representing the normalized span (b/b_{ref}) and x_2 the normalized root chord ($c_r/c_{r,ref}$). The reference values used for normalization are the baseline parameters of the C-130 model. **SSM** stands for **Static Stability Margin**,it is the Static Margin with opposite sign, to ensure longitudinal stability it has been constrained to be greater than or equal to 0.25. ν_h stands for **horizontal tailplane volumetric coefficient**, it has been constrained to reamain near 1, because the horizontal plane is fixed, indeed to ensure that the horizontal plane is large enough in relation to the wing to maneuver the aircraft, this coefficient must remain around 1.

The optimization was performed in two configurations: unconstrained and constrained. For benchmarking purposes, both the custom algorithm and the SciPy *SLSQP* solver were employed. The results are summarized in Table 1.1 and visualized in Figure 1.2.

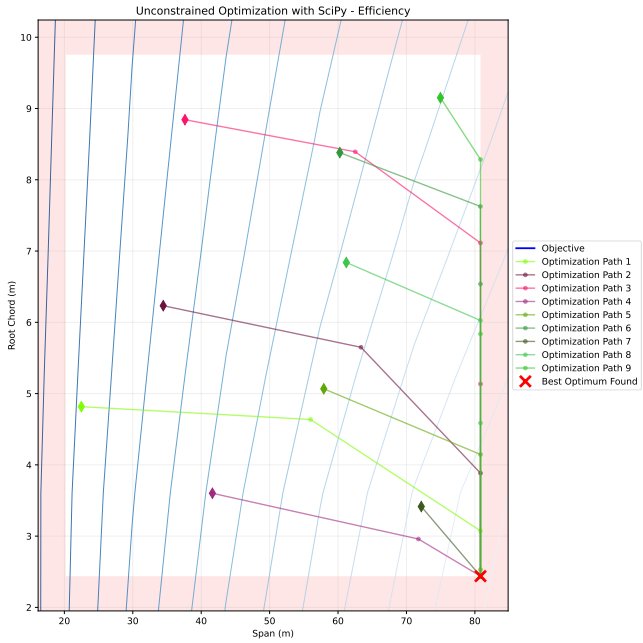
In the **unconstrained case** (Fig. 1.2a and 1.2b), the solver correctly identifies that to maximize aerodynamic efficiency, the aspect ratio must be maximized. Consequently, the optimization path pushes the span to the upper bound and the chord to the lower bound. Both solvers converged to the same region, with SciPy being faster due to the simplicity of the solution space.

In the **constrained case** (Fig. 1.2c and 1.2d), the feasible region is limited by the stability constraints. The optimal solution is no longer at the corner of the box but lies on the boundary.

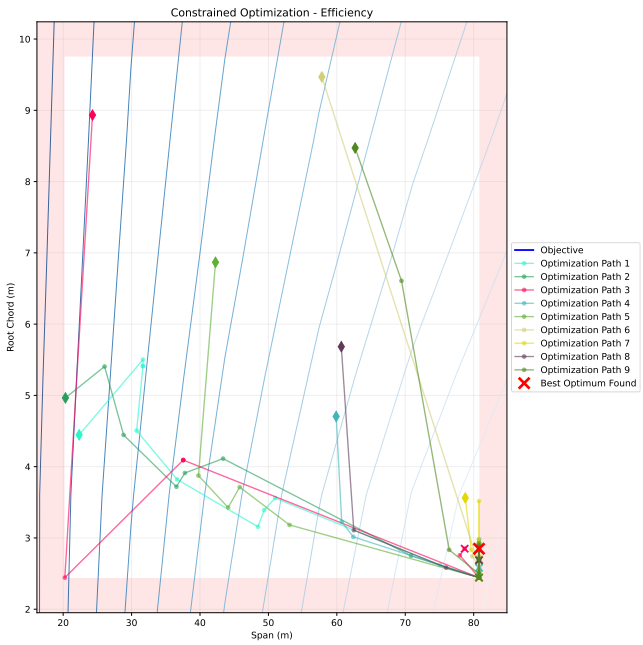
As shown in Table 1.1, the custom algorithm demonstrated superior efficiency compared to SciPy, requiring significantly fewer function evaluations and less computation time to reach a comparable optimum. This suggests that the specific implementation of the penalty method and line search in the custom algorithm is well-suited for this particular non-linear design space.



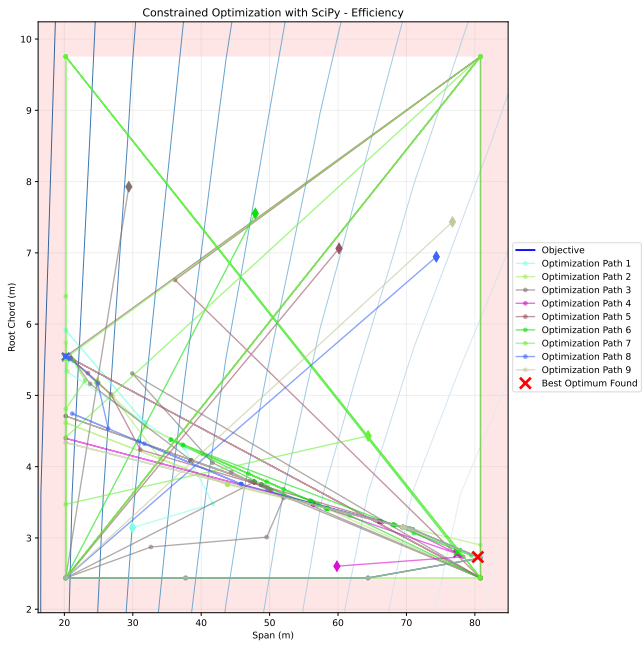
(a) Unconstrained (Custom)



(b) Unconstrained (SciPy)



(c) Constrained (Custom)



(d) Constrained (SciPy)

Figure 1.2: Optimization paths for aerodynamic efficiency. The red areas indicate the infeasible regions or bounds.

Case	Span [m]	Chord [m]	Efficiency (E)	SSM	ν_h
Custom Unconstrained	80.78	2.44	31.97	0.320	1.50
SciPy Unconstrained	80.82	2.44	31.44	0.322	1.50
Custom Constrained	80.78	2.85	31.67	0.225	1.09
SciPy Constrained	80.43	2.73	31.66	0.250	1.2
Case	**Time [s]**	**Iter**	**Func Evals**		
Custom Unconstrained	98.44	21	228		
SciPy Unconstrained	19.61	34	102		
Custom Constrained	2083.57	89	1745		
SciPy Constrained	9652.50	715	7458		

Table 1.1: Comparison of optimization results for Aerodynamic Efficiency.

1.2.2 Weight Estimation Discipline - MTOW Optimization

The second disciplinary optimization focuses on the aircraft weight, specifically the **Maximum Take-Off Weight** (MTOW). The weight estimation module relies on semi-empirical formulas that depend iteratively on the total weight itself. Consequently, the objective function evaluation involves an internal iterative loop to ensure consistency between the assumed weight and the calculated weight. The optimization problem is formulated as:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) = \frac{W_{TO}(x)}{W_{ref}} \\ \text{subject to} \quad & g_1(x) = \frac{0.25 - SSM(x)}{SSM_{ref}} \leq 0 \\ & g_2(x) = \nu_h(x) - 1.1 \leq 0 \\ & g_3(x) = 0.8 - \nu_h(x) \leq 0 \\ & x_{lower} \leq x \leq x_{upper} \end{aligned} \tag{1.1}$$

where W_{TO} is the converged take-off weight and $W_{ref} = 79,000$ kg is the baseline reference weight used for normalization. The calculation of W_{TO} requires a nested convergence loop:

$$W_{new} = \mathcal{F}(W_{old}, \text{Geometry}) \tag{1.2}$$

This loop continues until $|W_{new} - W_{old}| \leq \epsilon$, where $\epsilon = 0.1$ kg. Only after this internal convergence is achieved, the objective function value is returned to the optimizer. This optimization task was performed using the custom gradient-based algorithm in both unconstrained and constrained configurations. The results are summarized in Table 1.2 and the optimization paths are visualized in Figure 1.3.

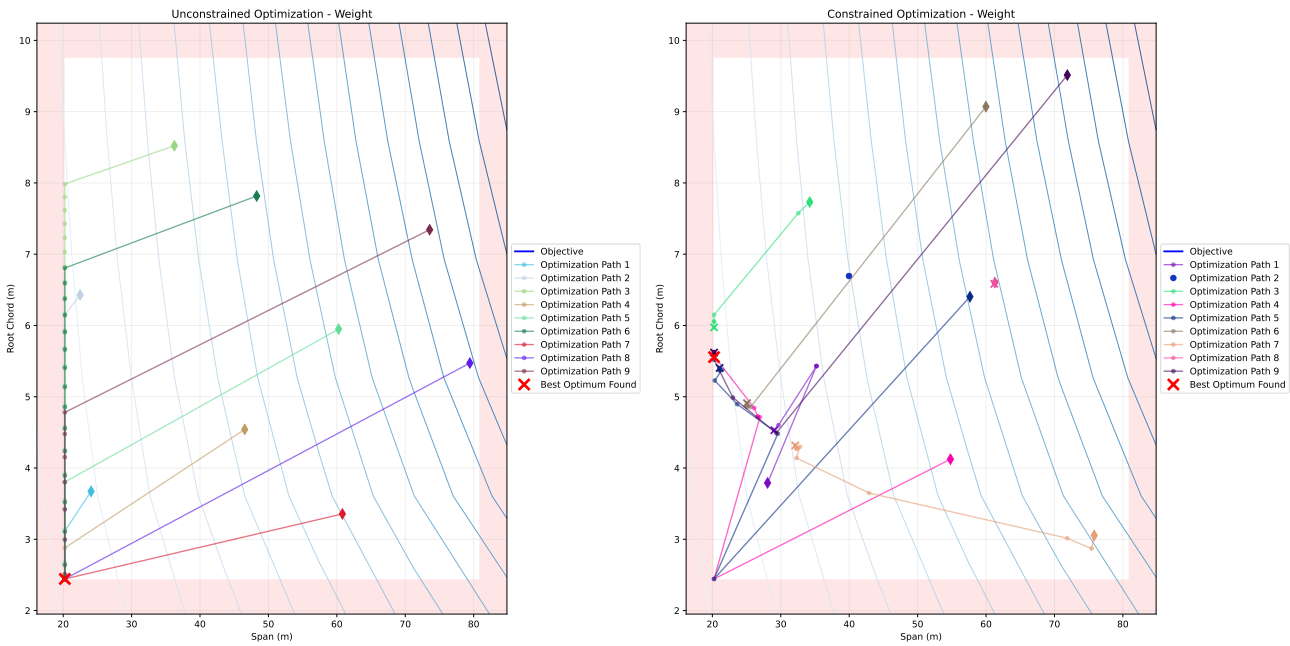
In the **unconstrained case** (Fig. 1.3a), the optimizer correctly identifies that the structural weight is minimized by reducing the aircraft's physical dimensions. Consequently, the solution converges to the lower bounds of the design space, resulting in a significantly lower MTOW. However, this configuration is physically unfeasible for a transport aircraft due to the lack of stability and lifting surface area.

In the **constrained case** (Fig. 1.3b), the algorithm must balance the weight reduction with the stability requirements. The span is still minimized to reduce the structural weight of the wing, but the root chord is significantly increased compared to the unconstrained case. This increase in chord length is necessary to satisfy the constraints.

Regarding computational performance, the presence of the internal weight convergence loop notably increased the computational time compared to the simple aerodynamic optimization.

Case	Span [m]	Chord [m]	MTOW [kg]	SSM	ν_h
Unconstrained	20.25	2.44	57132	1.04	5.97
Constrained	20.25	5.56	57908	0.25	1.09
Case	**Time [s]**	**Iter**	**Func Evals**		
Unconstrained	1948.70	78	1959		
Constrained	2296.10	38	1712		

Table 1.2: Comparison of optimization results for Weight Estimation (MTOW).



(a) Unconstrained Optimization (b) Constrained Optimization

Figure 1.3: Optimization paths for Maximum Take-Off Weight (MTOW).

1.2.3 Performance Discipline Optimization: STO, fuel, time

The performance analysis is critical as it directly dictates the operational capabilities of the aircraft. The optimization studies presented here utilize the Excel-based performance module to evaluate metrics such as **Take-Off Distance**, **Mission Time**, and **Total Fuel Consumption**.

Take-Off Distance Optimization

The optimization problem aims to minimize S_{TO} subject to the standard stability constraints and with the same design variables. The problem is formulated as follows:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) = \frac{S_{TO}(x)}{S_{ref}} \\ \text{subject to} \quad & g_1(x) = \frac{0.25 - SSM(x)}{SSM_{ref}} \leq 0 \\ & g_2(x) = \nu_h(x) - 1.1 \leq 0 \\ & g_3(x) = 0.8 - \nu_h(x) \leq 0 \\ & x_{lower} \leq x \leq x_{upper} \end{aligned} \tag{1.3}$$

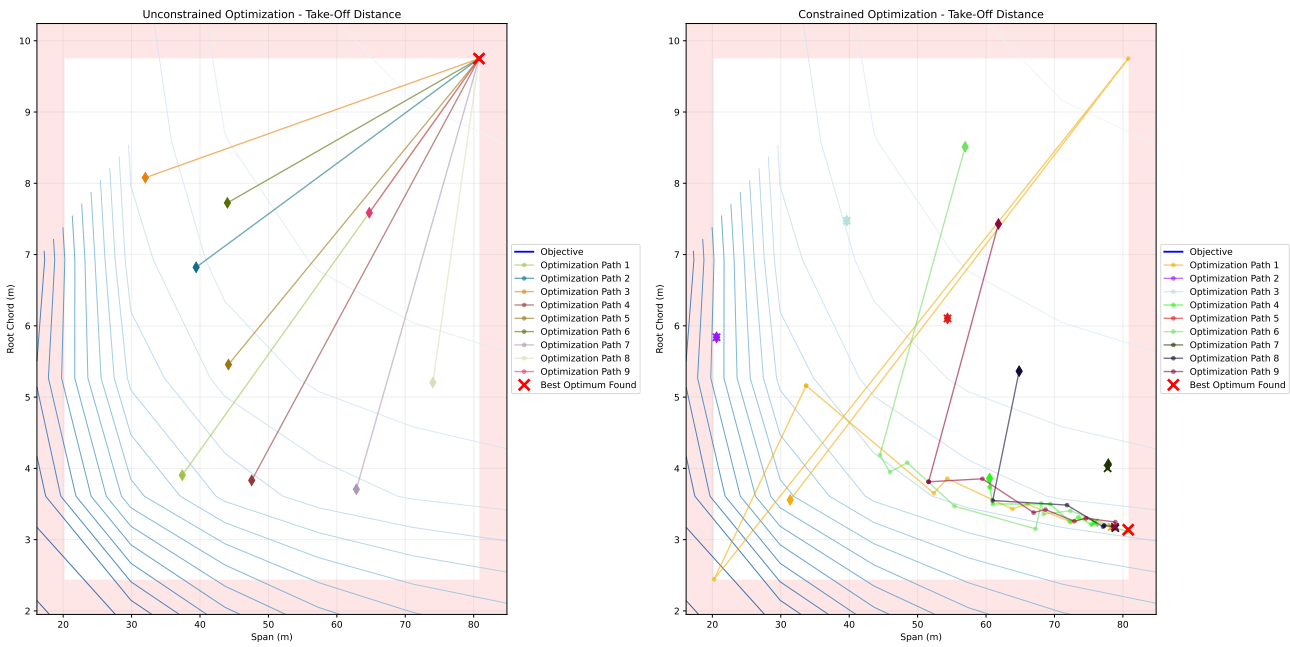
where S_{ref} is the reference take-off distance used for normalization, that is the C-130 real wing surface. The results are detailed in Table 1.3 and are shown in Figure 1.4.

In the **unconstrained case** (Fig. 1.4a), the optimizer aims to maximize the lifting capability of the aircraft to reduce the take-off run. Since the lift is proportional to the wing area (S) and the lift coefficient (C_L), the algorithm pushes both the span and the chord towards their upper bounds. The resulting configuration features a massive wing, which drastically reduces the take-off distance. However, this configuration is statically unstable ($SSM < 0$).

In the **constrained case** (Fig. 1.4b), the optimization is heavily restricted by the stability requirements. While the solver still attempts to maximize the span to reduce induced drag and increase lift, the root chord is constrained to a much lower value compared to the unconstrained case. The resulting take-off distance is approximately more than double the unconstrained optimum. It can be seen both in the contour plot of the objective and in the optimization paths in the constrained case that at low aspect ratios the maximum takeoff distance cannot be calculated because the aircraft is unable to perform the mission, so the optimization stops due to an error when the next step would have been in an area where the function is not defined.

Case	**Span [m]**	**Chord [m]**	S_{TO} **[m]**	**SSM**	ν_h
Unconstrained	80.78	9.75	457.56	-0.017	0.08
Constrained	80.8	3.14	913,64	0.175	0.89
Case	**Time [s]**	**Iter**	**Func Evals**		
Unconstrained	131.80	12	131		
Constrained	1845.52	48	1553		

Table 1.3: Comparison of optimization results for Take-Off Distance (S_{TO}).



(a) Unconstrained Optimization

(b) Constrained Optimization

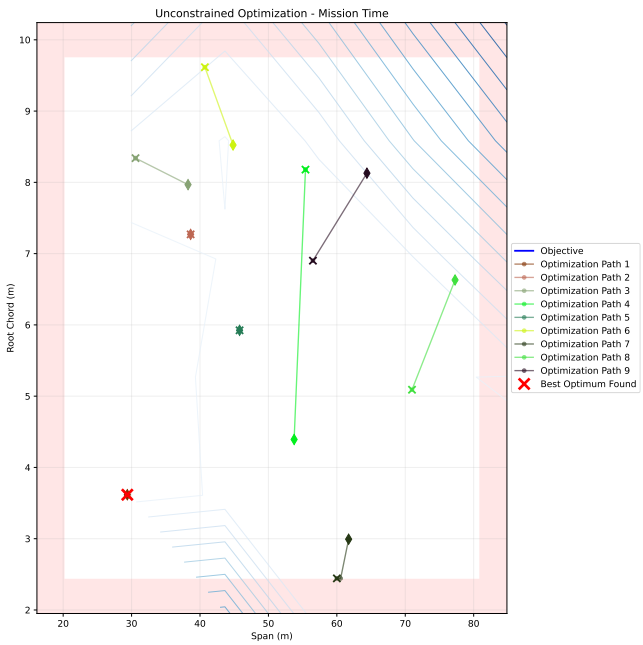
Figure 1.4: Optimization paths for Take-Off Distance.

Mission Time Optimization

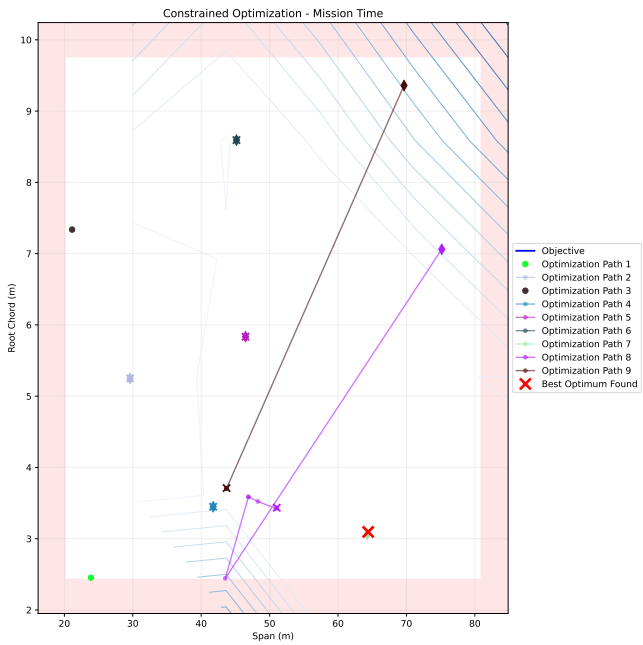
The second performance metric analyzed is the total Mission Time ($t_{mission}$) calculated by the performance module. The formulation follows the standard structure used in the previously:

$$
\begin{aligned}
\underset{x}{\text{minimize}} \quad & f(x) = \frac{t_{mission}(x)}{t_{ref}} \\
\text{subject to} \quad & g_1(x) = \frac{0.25 - SSM(x)}{SSM_{ref}} \leq 0 \\
& g_2(x) = \nu_h(x) - 1.1 \leq 0 \\
& g_3(x) = 0.8 - \nu_h(x) \leq 0 \\
& x_{lower} \leq x \leq x_{upper}
\end{aligned} \tag{1.4}
$$

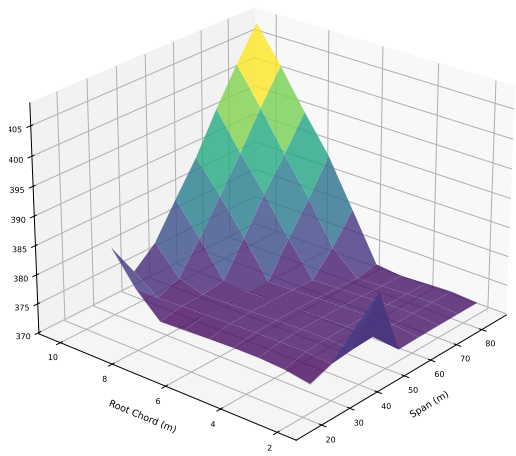
The optimization results for the mission time present a distinct behavior compared to the previous disciplines. As illustrated in the 3D surface plot (Figure 1.5c), the objective function is characterized by a vast, nearly flat region covering the majority of the design space. Significant variations in mission time are observed only at the extreme boundaries of the domain. Consequently, the optimization algorithm tends to terminate as soon as it enters this flat region, as the gradient of the objective function becomes negligible ($\nabla f \approx 0$). The numerical results are summarized in Table 1.4, and the optimization paths are visualized in Figure 1.5.



(a) Unconstrained Optimization Path



(b) Constrained Optimization Path



(c) 3D Visualization of the Mission Time

Figure 1.5: Optimization results for Mission Time.

Case	Span [m]	Chord [m]	Time [min]	SSM	ν_h
Unconstrained	29.37	3.62	371.5	0.36	1.84
Constrained	64.40	3.10	372.0	0.23	1.16
Case	**Time [s]**	**Iter**	**Func Evals**		
Custom Unconstrained	1335.86	10	1778		
Custom Constrained	143.30	9	159		

Table 1.4: Comparison of optimization results for Mission Time.

Mission Fuel Consumption Optimization

The objective is to minimize the total fuel mass estimated by the performance module. The reference fuel mass $m_{fuel,ref}$ is used for normalization. Unlike the previous cases, preliminary tests indicated that the original stability constraints were too restrictive for the fuel optimization landscape, hindering the algorithm's convergence. Consequently, the constraints were slightly relaxed to enlarge the feasible design space.

The optimization problem is formally stated as:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) = \frac{m_{fuel}(x)}{m_{fuel,ref}} \\ \text{subject to} \quad & g_1(x) = \frac{0.15 - SSM(x)}{SSM_{ref}} \leq 0 \\ & g_2(x) = \nu_h(x) - 1.25 \leq 0 \\ & g_3(x) = 0.75 - \nu_h(x) \leq 0 \\ & x_{lower} \leq x \leq x_{upper} \end{aligned} \tag{1.5}$$

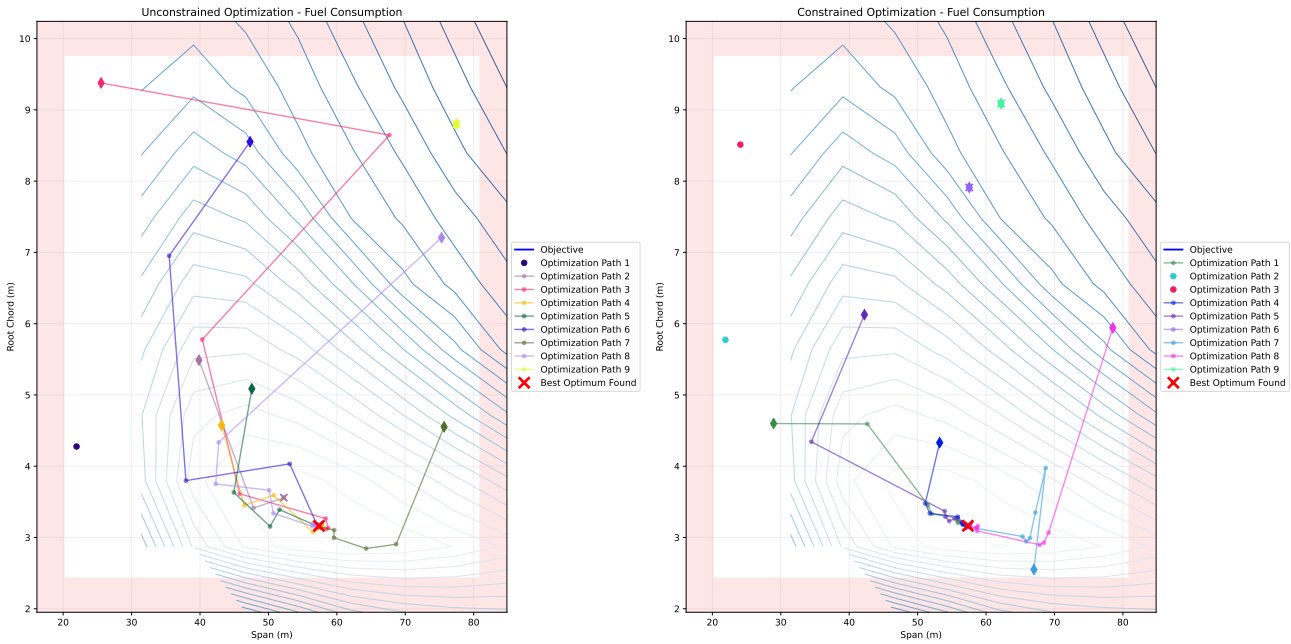
The results of the optimization using the custom algorithm are presented in Table 1.5 and visualized in Figure 1.6. A key observation from this analysis is the convergence of both the unconstrained and constrained cases to practically the same solution. This specific geometry represents a trade-off between aerodynamic efficiency and structural weight:

- Increasing the span further (as seen in the aerodynamic efficiency case) would reduce induced drag but significantly increase the structural weight, thereby requiring more fuel to carry the extra mass.
- Reducing the span (as seen in the MTOW case) would reduce structural weight but degrade aerodynamic performance, requiring more thrust and thus more fuel.

Since this trade-off point naturally results in an SSM of ≈ 0.24 and an ν_h of ≈ 1.24, the solution satisfies the relaxed constraints imposed in this study. Consequently, the unconstrained optimum lies within the feasible region, explaining why the results are nearly identical. It is important to note that this objective is not defined for spans less than approximately 32 m. For this reason, it was decided to restrict the lower bounds of the problem to a span of 34 m and a root chord of 3.4 m.

Table 1.5: Comparison of optimization results for Mission Fuel Consumption.

Case	Span [m]	Chord [m]	Fuel [kg]	SSM	ν_h
Unconstrained	57.38	3.16	13086.6	0.242	1.24
Constrained	57.34	3.16	13086.6	0.242	1.24
Case	**Time [s]**	**Iter**	**Func Evals**		
Custom Unconstrained	516.44	49	605		
Custom Constrained	874.72	50	801		



(a) Unconstrained Optimization Path

(b) Constrained Optimization Path

Figure 1.6: Optimization paths for Mission Fuel Consumption.

1.2.4 Structural Discipline Formulation

In the previous single-discipline optimizations, the solver tended to maximize the wing span to reduce induced drag, leading to aerodynamically efficient but structurally unfeasible designs. To mitigate this issue without integrating a computationally expensive Finite Element Analysis (FEA) loop, a simplified structural objective function was introduced. The fundamental structural requirement for a cantilever wing is to resist the root bending moment, which increases with the wing span. To maintain structural integrity comparable to the baseline aircraft, the wing root section must possess an adequate **Area Moment of Inertia** (I). A geometric proxy was defined to capture the relationship between the wing's bending stiffness and its span. The root section is approximated as a rectangular beam with width equal to the root chord (c_r) and height equal to the maximum thickness of the airfoil ($t_{max} = c_r \cdot (t/c)_{max}$). The Area Moment of Inertia is thus estimated as:

$$I_{root} \approx \frac{1}{12} c_r (c_r \cdot (t/c)_{max})^3 \tag{1.6}$$

To account for the increased bending moment imposed by a longer wing, a *Stiffness-to-Span ratio* ($\mathcal{K}$) was introduced as the performance metric:

$$\mathcal{K} = \frac{I_{root}}{b/2} \tag{1.7}$$

The design intent is to penalize deviations from the baseline aircraft's structural characteristics. Therefore, the optimization seeks to minimize the relative difference between the current ratio $\mathcal{K}$ and the reference ratio $\mathcal{K}_{ref}$.

Initially, a linear formulation of the objective function was tested:

$$f_{struct,linear} = \left| \frac{\mathcal{K}}{\mathcal{K}_{ref}} - 1 \right| \tag{1.8}$$

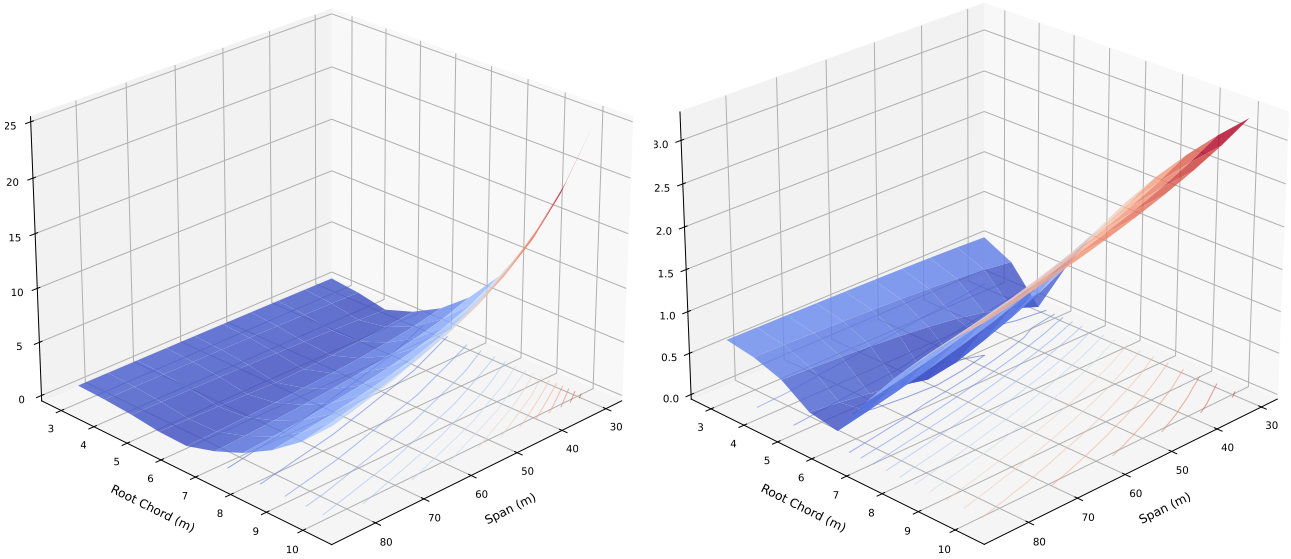
However, an analysis of the design space revealed that this function produces values ranging from 0 to 25 (Fig. 1.7a). Such a large range and steep gradients at the domain boundaries

can overshadow other disciplines in a Multi-Disciplinary Design Optimization (MDO) context, causing the optimizer to focus exclusively on the structural constraint.

To address this, a **logarithmic scaling** was applied. The final structural objective function is defined as:

$$f_{struct} = \log_{10}\left(\left|\frac{\mathcal{K}}{\mathcal{K}_{ref}} - 1\right| + 1\right) \tag{1.9}$$

This transformation compresses the high-penalty regions while preserving the sensitivity near the optimum (where the difference approaches zero), as shown in Figure 1.7b. This ensures a smoother gradient landscape, improving the convergence behavior of the optimizer when coupled with other disciplines.



(a) Linear Formulation (max value = 25)

(b) Logarithmic Formulation (max value = 3)

Figure 1.7: Comparison of the structural objective function landscapes.

1.3 Multi-Disciplinary Design Optimization (MDO)

Before presenting the Multi-Disciplinary Design Optimization (MDO) results, it is important to note a modification in the baseline aircraft configuration used for this section. The wing position along the fuselage has been shifted from 13 m to 10 m from the nose. This geometric update alters the aircraft's center of gravity and aerodynamic center lever arms, resulting in slightly different stability characteristics and trim drag values compared to the single-discipline analyses presented in the previous sections.

1.3.1 Direct Optimization with Composite Objective Function

The first MDO approach involves a direct optimization of the coupled system. The goal is to simultaneously improve conflicting performance metrics: Fuel Consumption, Take-Off Distance, Mission Time, MTOW, Aerodynamic Efficiency and structural stiffness while satisfying stability constraints.

Optimization Architecture (XDSM)

The data flow and the coupling between the different disciplines are visualized in the **Extended Design Structure Matrix** (XDSM) in Figure 1.8. The process begins with the optimizer defining the geometry (x). The *Geometry* module updates the aircraft configuration, which is then fed into the *Aerodynamics* and *Structures* modules. A dedicated convergence loop (orange block) ensures consistency between the estimated weight and the calculated Maximum Take-Off Weight (MTOW). Finally, the *Performance* module evaluates the mission metrics, which are aggregated into a single objective function.

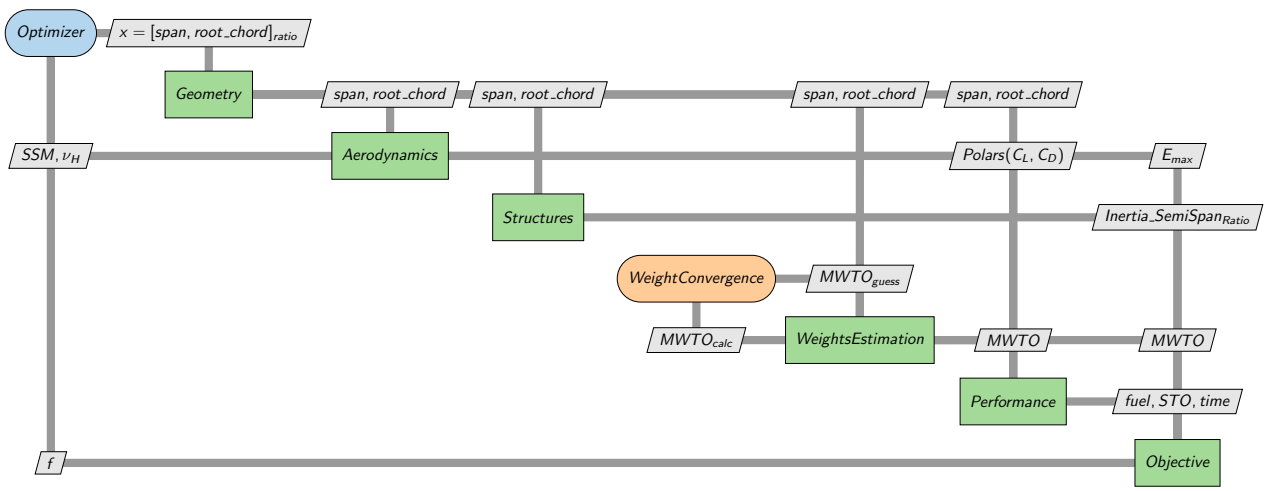


Figure 1.8: XDSM diagram of the Multi-Disciplinary Design Optimization process.

Problem Formulation

To handle the multi-objective nature of the problem using a single-objective optimizer, a *Weighted Sum Method* was employed. All objectives were normalized by their respective reference values (denoted by the superscript $(\cdot)^*$) to obtain dimensionless quantities of comparable magnitude.

The optimization problem is formally stated as follows:

$$
\begin{aligned}
\underset{x}{\text{minimize}} \quad & J(x) = \frac{1}{4}\left(\frac{m_{fuel}(x)}{m^*_{fuel}} + \frac{S_{TO}(x)}{S^*_{TO}} + \frac{t_{mission}(x)}{t^*_{mission}} + \frac{W_{TO}(x)}{W^*_{TO}} - \frac{E(x)}{E^*} + f_{struct}(x)\right) \\
\text{subject to} \quad & g_1(x) = \frac{0.15 - SSM(x)}{SSM_{ref}} \leq 0 \\
& g_2(x) = \nu_h(x) - 1.25 \leq 0 \\
& g_3(x) = 0.75 - \nu_h(x) \leq 0 \\
& x_{lower} \leq x \leq x_{upper}
\end{aligned}
\tag{1.10}
$$

where the efficiency term E/E^* is subtracted (since the goal is maximization), while all other terms are added (minimization). The factor 1/4 serves as a scaling constant.

Results and Discussion

The optimization was performed for both unconstrained and constrained cases. The results are summarized in Table 1.6 and the optimization paths are shown in Figure 1.9.

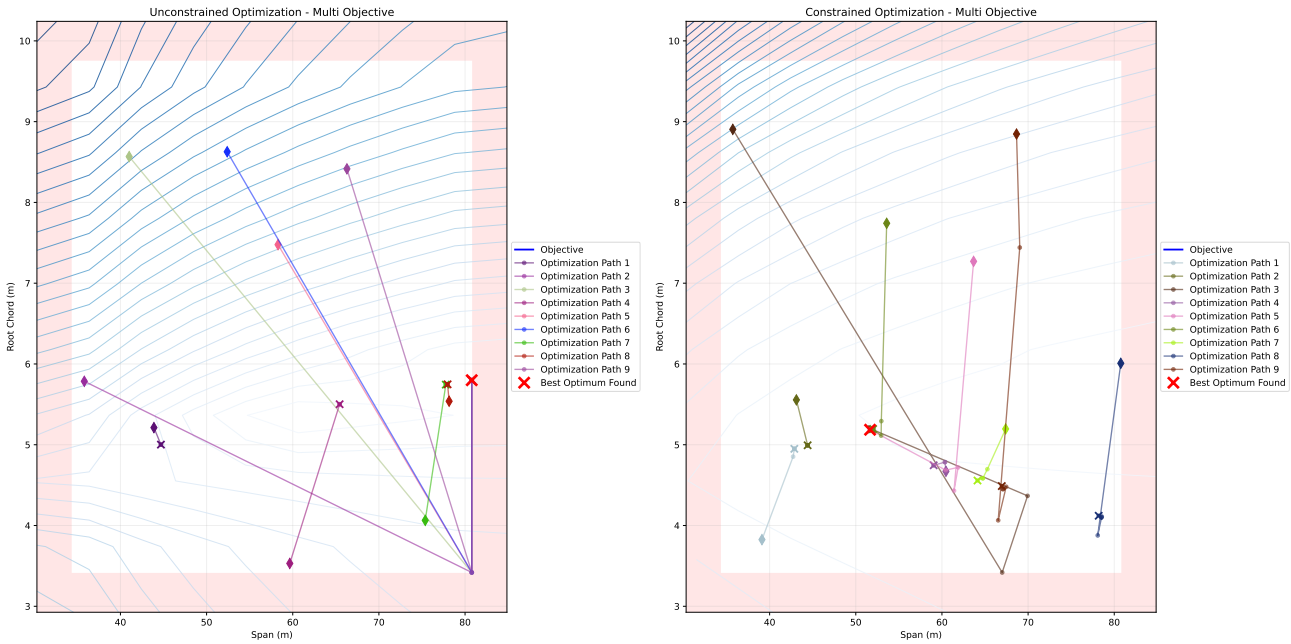
In the unconstrained case (Fig. 1.9a), the solver pushed the design towards the upper bound of the span to maximize aerodynamic efficiency (E), which heavily influences fuel consumption. However, the structural penalty (f_{struct}) prevented the chord from minimizing excessively, resulting in a large chord to maintain inertia.

In the constrained case (Fig. 1.9b), the stability requirements significantly limited the design space. The optimal span was reduced to 51.66 m, and the chord settled at 5.19 m. This configuration represents a balanced compromise: the Take-Off distance (2875 m) is higher than the unconstrained theoretical minimum, but the solution ensures static stability ($SSM = 0.27$) and controllability ($\nu_h = 0.68$).

Parameter	**Unconstrained**	**Constrained**
Span [m]	80.77	51.66
Root Chord [m]	5.80	5.19
Objective Function $J(x)$	0.4066	0.6998
Total Fuel [kg]	16546	14078
Take-Off Distance [m]	769.68	1035
Mission Time [min]	371.95	372.50
MTOW [kg]	74836	65575
SSM	0.054	0.267
Execution Time [s]	542	2638.48

Table 1.6: Results of the Direct Multi-Disciplinary Optimization.

A critical observation concerns the **computational cost**. The constrained optimization required approximately 44 minutes (2638 s) to converge to a single optimal point, involving over 1200 function evaluations. The next phase of the study aims to generate a Pareto Front to analyze the trade-offs between objectives. Constructing a high-resolution Pareto front typically requires finding tens or hundreds of optimal points. Using this direct optimization approach would be computationally prohibitive (estimated 36 hours for 50 points). Therefore, to enable efficient Pareto analysis, the strategy will shift to the use of surrogate models (Response Surfaces) to approximate the discipline outputs, drastically reducing the evaluation time.



(a) Unconstrained MDO Path

(b) Constrained MDO Path

Figure 1.9: Optimization paths for the Multi-Disciplinary case.

1.3.2 Response Surface Methodology (RSM)

Given the high computational cost observed in the direct constrained optimization (approximately 44 minutes for a single optimal point), generating a high-resolution Pareto front using the original Excel-based model would be prohibitively time-consuming. To overcome this limitation, a surrogate modeling approach based on Response Surface Methodology (RSM) was implemented. This technique allows for the replacement of the computationally expensive disciplinary analyses with fast-evaluating mathematical approximations.

Design of Experiments and Data Acquisition

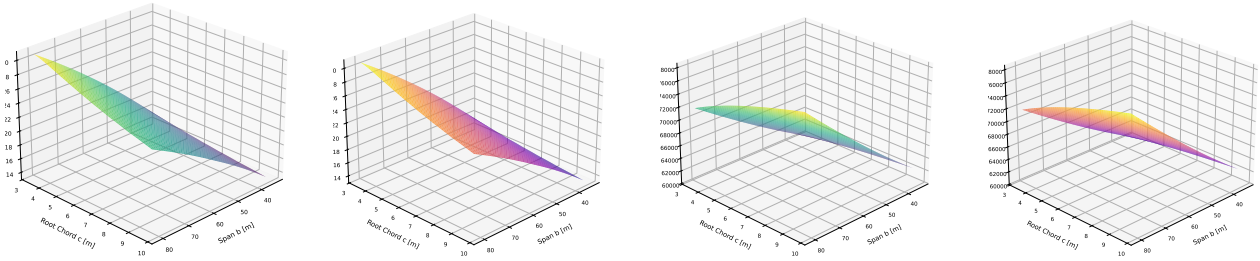
The first step in constructing the surrogate models involved the exploration of the design space to gather a comprehensive dataset. A Full Factorial Design of Experiments (DOE) was established by discretizing the domain into a regular grid. The design variables were sampled within the following bounds:

- Span Ratio (x_1): $[0.85, 2.0]$
- Root Chord Ratio (x_2): $[0.7, 2.0]$

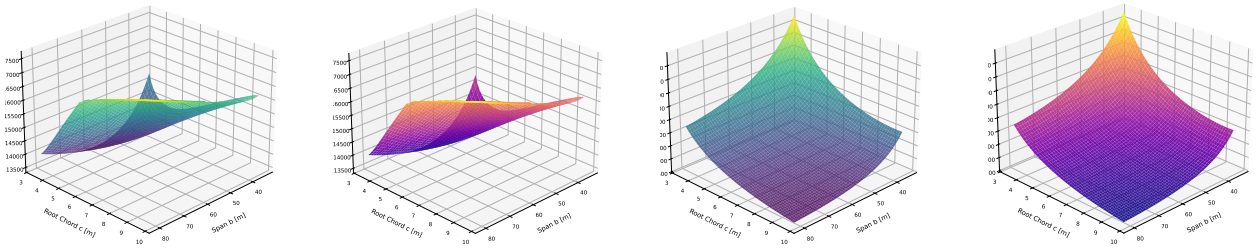
A grid resolution of 50×50 points was selected, resulting in a total of 2500 function evaluations. Based on the reference aircraft geometry ($b_{ref} = 40.41$ m and $c_{root,ref} = 4.877$ m), this discretization provides a physical sampling accuracy of approximately $\Delta b \approx 0.95$ m for the wing span and $\Delta c \approx 0.13$ m for the root chord. This density was deemed sufficient to capture the non-linearities of the physical phenomena without incurring excessive computational time for the dataset generation.

Once the dataset was generated, the surrogate models were constructed using interpolation techniques. Since the data was structured on a regular rectangular grid, Bivariate Spline interpolation was selected as the most appropriate mathematical method. Specifically, the ‘RectBivariateSpline‘ function from the *SciPy* library was utilized. A cubic spline interpolation ($k_x = 3, k_y = 3$) was applied to ensure C^2 continuity (continuity of the function, its first,

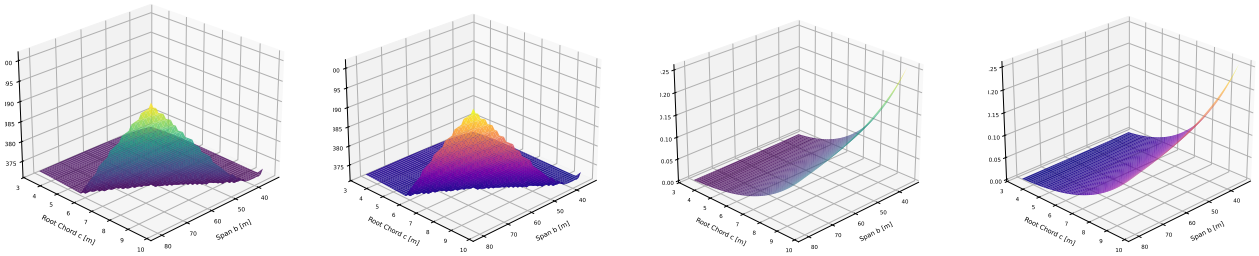
and second derivatives) across the design space. This property is crucial for gradient-based optimization algorithms, which rely on smooth derivatives to converge efficiently. The cubic splines demonstrated a smooth representation of the disciplinary physics, faithfully reproducing the trends observed in the direct (Figure 1.10).



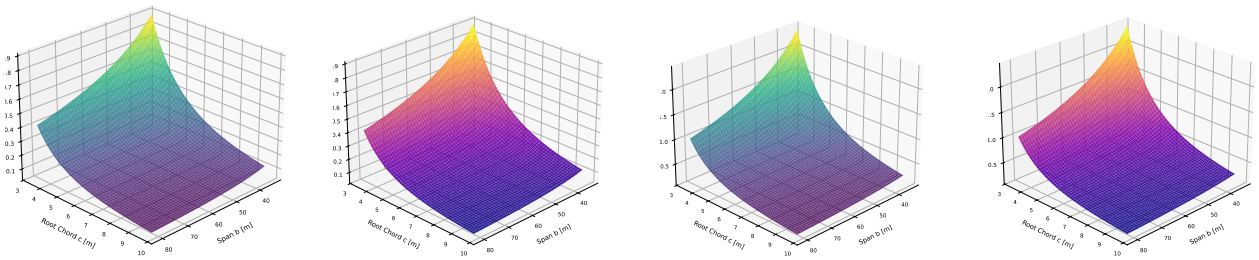
(a) E (Excel) (b) E (Spline) (c) MTOW (Excel) (d) MTOW (Spline)



(e) Fuel (Excel) (f) Fuel (Spline) (g) S_{TO} (Excel) (h) S_{TO} (Spline)



(i) Time (Excel) (j) Time (Spline) (k) Inertia R. (Excel) (l) Inertia R. (Spline)



(m) SSM (Excel) (n) SSM (Spline) (o) ν_h (Excel) (p) ν_h (Spline)

Figure 1.10: Validation of the Response Surface Models..

1.3.3 Optimization with Surrogate Models

Following the construction and verification of the Response Surface Models (RSM), the Multi-Disciplinary Design Optimization was re-executed replacing the Excel analysis with the spline interpolators.

Modified Optimization Architecture

The updated data flow is depicted in the XDSM diagram in Figure 1.11. The logical structure of the optimization remains identical to the direct approach: the optimizer iteratively updates the design variables (x) to minimize the same composite objective function $J(x)$ subject to the same stability constraints. However, the external calls are replaced by instantaneous evaluations of the loaded spline objects.

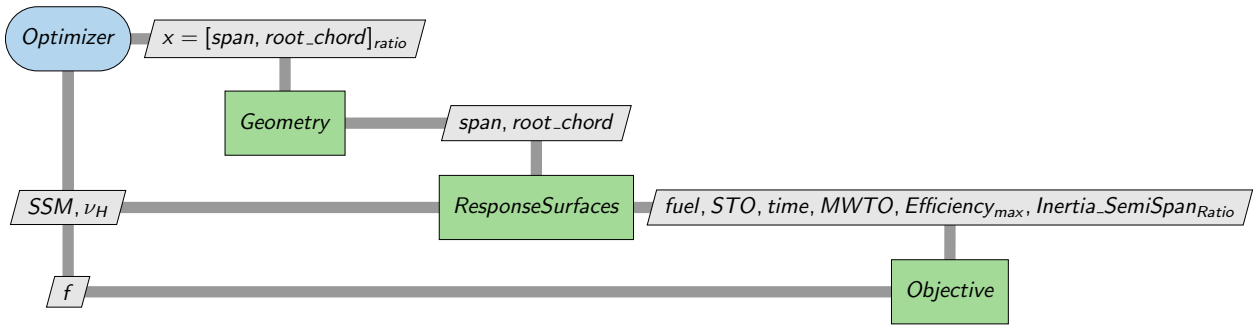


Figure 1.11: XDSM diagram of the MDO process utilizing Surrogate Models (Response Surfaces).

Results Validation and Comparison

The results of the surrogate-based optimization are presented in Table 1.7 and visualized in Figure 1.12. Comparing these results with those obtained from the direct optimization, a strong correlation is observed.

In the **unconstrained case** the surrogate optimizer converged to a span of 80.77 m and a chord of 5.80 m. These values are virtually identical to the direct method results (80.77 m and 5.80 m), confirming that the splines accurately capture the global gradient trends, particularly the drive towards high aspect ratio for aerodynamic efficiency.

In the **constrained case** the optimization settled at a span of 49.15 m and a root chord of 5.12 m. This solution is geometrically very close to the direct optimization optimum (51.66 m and 5.19 m). The slight deviation ($< 5\%$ in span) is attributable to the smoothing effect of the cubic splines near the sharp boundaries of the stability constraints.

Regarding **computational cost** the results in Table 1.7 demonstrate a drastic improvement in performance. The constrained optimization time dropped from 2638.48 seconds (approx. 44 minutes) to just 0.12 seconds. This represents a speedup factor of approximately order 10^4.

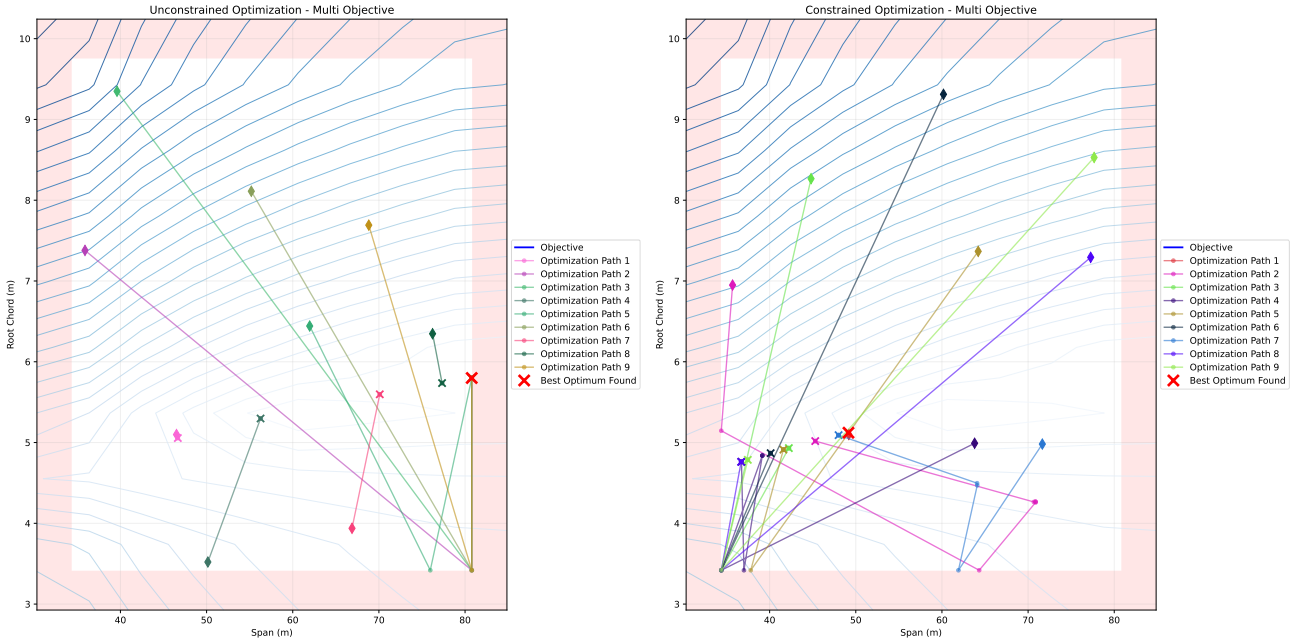
This massive reduction in execution time makes the generation of a high-resolution Pareto Front feasible. Recalling the previous estimation, generating 50 optimal points for a Pareto front using the direct method would have required over 30 hours. With the surrogate models, the same task is estimated to take:

$$T_{Pareto} \approx 50 \times 0.12 \text{ s} = 6 \text{ seconds} \tag{1.11}$$

Consequently, the study can proceed to the multi-objective trade-off analysis with negligible computational overhead.

	Unconstrained		Constrained	
Parameter	*Direct*	*Spline*	*Direct*	*Spline*
Span [m]	80.77	80.77	51.66	49.15
Root Chord [m]	5.80	5.80	5.19	5.12
Objective Function $J(x)$	0.4066	0.4063	0.6998	0.4780
Execution Time [s]	542	**0.06**	2638.48	**0.12**

Table 1.7: Comparison between Direct Optimization and Surrogate-Based Optimization results.



(a) Unconstrained Path (Surrogate)

(b) Constrained Path (Surrogate)

Figure 1.12: Optimization paths using the Surrogate Models.

1.3.4 Multi-Objective Pareto Analysis

The final stage of the optimization study aims to map the trade-offs between the six competing design objectives: Aerodynamic Efficiency, Structural Stiffness, MTOW, Fuel Consumption, Mission Time, and Take-Off Distance. Since no single design can simultaneously minimize all these metrics, the goal is to identify the Pareto Front: the set of non-dominated solutions where improving one objective inevitably degrades at least one other.

Given the high dimensionality of the objective space ($n = 6$), standard methods like the Weighted Sum Method often fail to produce an evenly distributed set of Pareto points, particularly if the front is non-convex. To overcome this, the **Normal Boundary Intersection (NBI)** method was implemented using the high-speed surrogate models.

Normal Boundary Intersection (NBI) Formulation

The NBI method constructs the Pareto front by finding the intersection between the boundary of the feasible objective space and a set of normal lines emanating from a convex hull defined by the individual single-objective optima.

The procedure implemented consists of the following steps:

1. **Anchor Points:** First, the individual minima for each of the $N = 6$ objectives are found. Let x_i^* be the minimizer of the i-th objective $f_i(x)$. The vector of these optimal values

constitutes the *Utopia Point* $f^* = [f_1(x_1^*), \ldots, f_N(x_N^*)]^T$.

2. **Pay-off Matrix:** A Convex Hull matrix P of size $N \times N$ is constructed, where the i-th column is given by $F(x_i^*) - f^*$.

3. **Quasi-Normal Vector:** A reference normal vector $\hat{n}$ is defined to point towards the unattainable region (towards the Utopia point), typically calculated as $\hat{n} = -\Phi \cdot \mathbf{1}$, where $\mathbf{1}$ is a column vector of ones.

The NBI method effectively transforms the multi-objective problem into a set of equality-constrained single-objective sub-problems. A set of weights β is generated such that $\sum \beta_i = 1$ and $\beta_i \geq 0$. To ensure a uniform coverage of the high-dimensional space, the weights were generated using a **Sobol sequence**.

For each weight vector β, the optimizer solves the following sub-problem:

$$\begin{aligned} \underset{x,t}{\text{maximize}} \quad & t \\ \text{subject to} \quad & P\beta + t\hat{n} = F(x) - f^* \\ & g(x) \leq 0 \quad \text{(Physical Constraints)} \end{aligned} \tag{1.12}$$

By maximizing the parameter t, the solver pushes the solution along the normal direction as far as possible until it hits the boundary of the feasible set, effectively locating a point on the Pareto front.

Pareto Front Results and Visualization

The NBI optimization was executed for both the unconstrained and constrained scenarios using the spline interpolators. The results are visualized using Scatter Plot in Figures 1.13 and 1.14. In these matrices:

- The **off-diagonal plots** represent the 2D projections of the Pareto front for each pair of objectives. These plots reveal the trade-offs (e.g., L-shaped or convex curves indicating conflicting goals). The star markers represent the Anchor Points (single-discipline optima).

- The **diagonal plots** display the histograms (frequency distribution) of the optimal solutions for each specific objective.

Figure 1.13 shows the results for the **unconstrained case**. The solutions are widely spread across the design space. Strong trade-offs are visible, for example, between *Efficency* and *WTO* or *Fuel Consumption* and *Take-Off Distance* (STO). Designs optimized for short take-off (requiring large wing areas) incur a significant drag and weight penalty, drastically increasing fuel consumption.

Figure 1.14 shows the results for the **constrained case**. The feasible design space is drastically reduced.

Comparing the histograms, the distribution of optimal solutions is much narrower than in the unconstrained case. The stability constraints force the majority of Pareto-optimal solutions to cluster around a specific geometric region (around 50m span). This indicates that while theoretical trade-offs exist, the requirement for a stable and controllable aircraft dominates the design, limiting the designer's freedom to extremely specialize the aircraft for a single metric (e.g., extreme STO or extreme Efficiency) without violating safety margins.

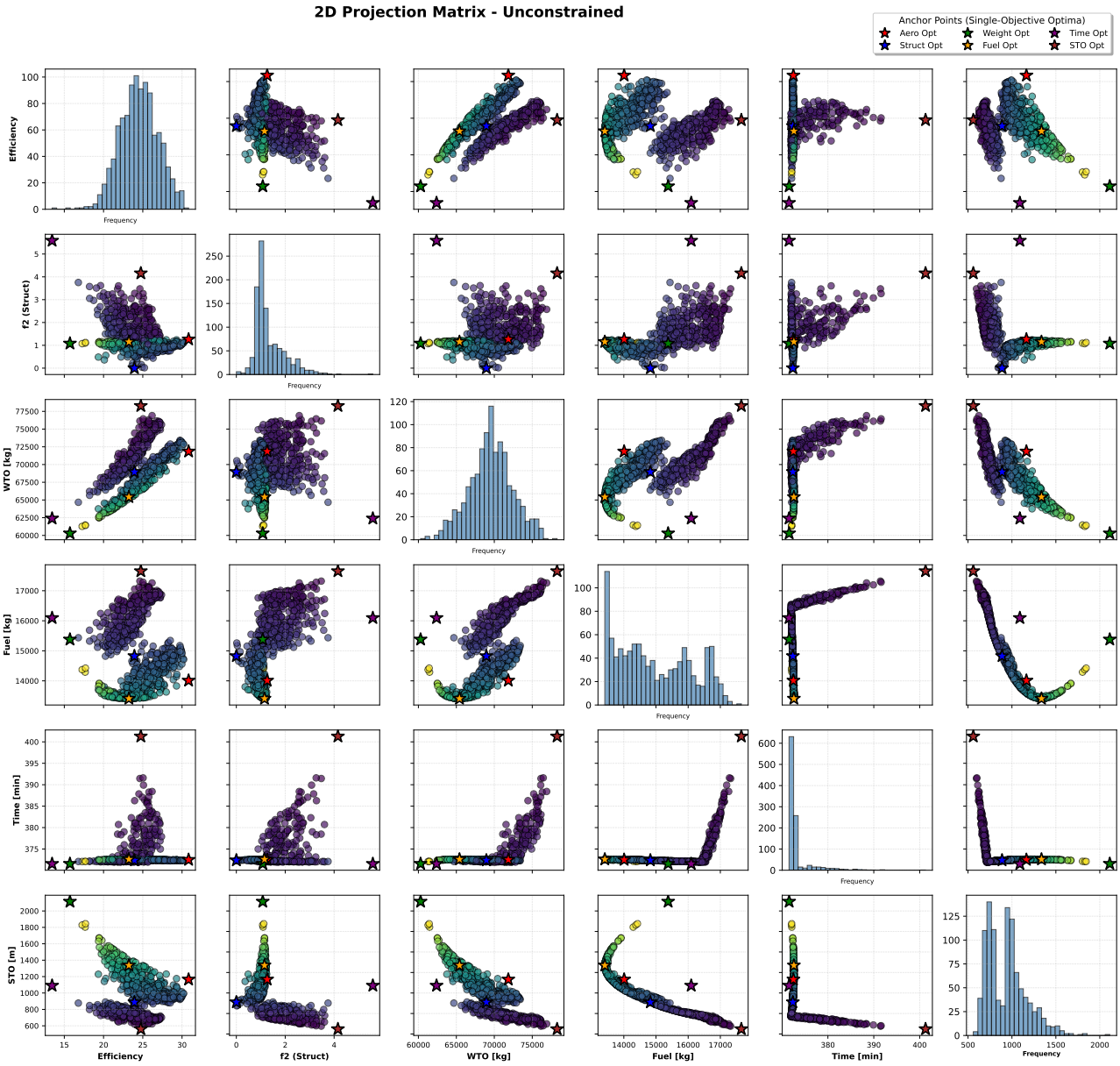


Figure 1.13: Scatter Plot Matrix of the Pareto Front for the Unconstrained case.

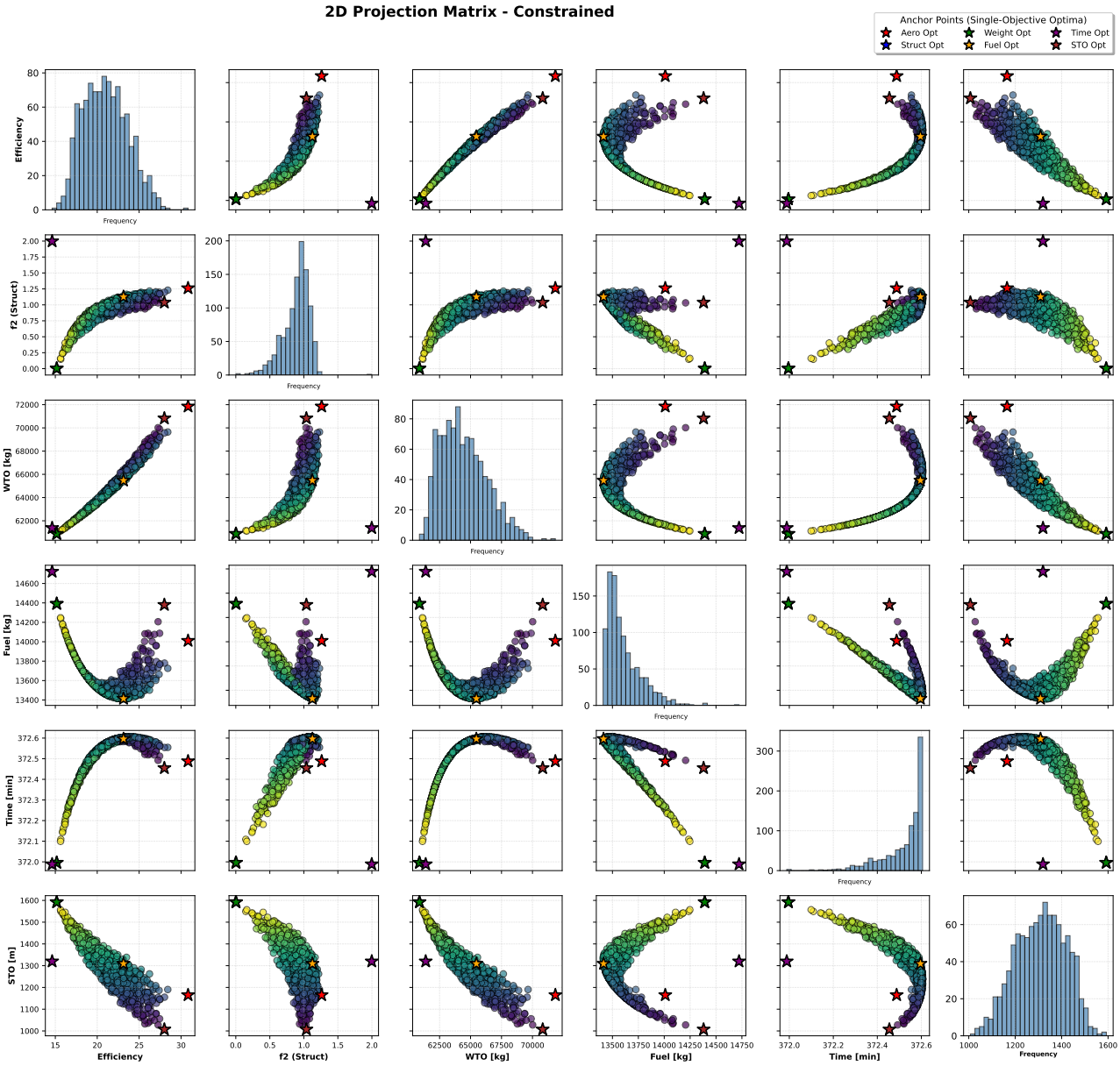


Figure 1.14: Scatter Plot Matrix of the Pareto Front for the Constrained case.

1.3.5 Pareto Front Analysis (3 Design Variables)

The final iteration of the optimization study expands the design space by introducing a third variable: the longitudinal position of the wing along the fuselage (X_{wing}). This variable is critical for finding the optimal balance between longitudinal static stability (Static Margin).

Expanded Response Surface Generation

Moving from a two-variable ($x \in \mathbb{R}^2$) to a three-variable ($x \in \mathbb{R}^3$) problem introduces the "curse of dimensionality," exponentially increasing the computational cost required to populate the design space. To maintain a feasible computational time for the dataset generation while ensuring sufficient accuracy, the grid resolution was adapted based on the physical sensitivity of the variables.

The expanded design space was bounded and sampled as follows:

- **Span (b):** Range $[0.85, 2.0] \cdot b_{ref}$ (approx. 46.5 m). Sampled with a step of ≈ 1.5 m.
- **Root Chord (c_r):** Range $[0.7, 2.0] \cdot c_{ref}$ (approx. 6.3 m). Sampled with a finer step of ≈ 0.25 m.
- **Wing Position (X_{wing}):** Range $[0.5, 1.7] \cdot X_{ref}$ (approx. 17.4 m). Sampled with a step of ≈ 1.5 m.

This adaptive sampling strategy resulted in a 3D grid containing approximately 10,000 evaluations. While the spatial resolution is coarser compared to the 2D study, the use of trivariate spline interpolation ensures that the global gradients and trends are adequately captured for the optimization algorithms.

Optimization Architecture

The updated MDO architecture is presented in Figure 1.15. The ‘Geometry‘ and ‘ResponseSurface‘ modules now accept the triplet vector $x = [b/b_{ref}, c_r/c_{ref}, X_{wing}/X_{ref}]^T$ as input. The constraints and objective function formulation remain consistent with the previous section ($J(x)$ composed of 6 terms), but the addition of X_{wing} provides the optimizer with a new degree of freedom to satisfy the stability constraints (SSM and ν_h) without necessarily penalizing the wing aspect ratio.

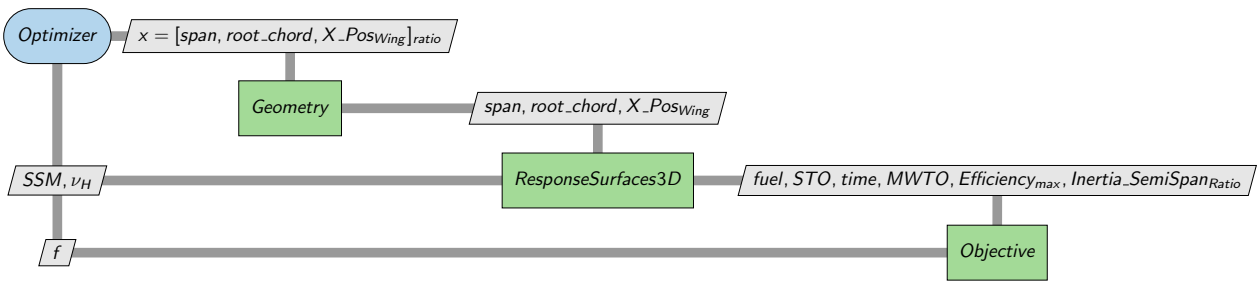


Figure 1.15: Updated XDSM diagram including Wing Position as a third design variable.

The NBI method was applied to the new 3D surrogate models to generate the Pareto fronts. The results for the unconstrained and constrained cases are visualized in the Scatter Plot Matrices in Figures 1.16 and 1.17.

In the **unconstrained case** (Fig. 1.16), the solution space is extremely broad. The optimizer exploits the freedom to move the wing to extreme positions to minimize trim drag,

often resulting in configurations that would be unflyable in reality (negative static margins). The histograms on the diagonal show a wide distribution of performance metrics, indicating that without stability limits, the "optimal" aircraft can take vastly different forms depending on which objective is prioritized like extremely long wings for efficiency vs. short wings for structural weight.

In the **constrained case** (Fig. 1.17) the constraints significantly prunes the design space. A comparison with the unconstrained matrix reveals a drastic reduction in the spread of the solutions. Notably, the ability to adjust the wing position allows the optimizer to satisfy the SSM and ν_H constraint more efficiently than in the fixed-wing case. Again, the feasible region remains tightly clustered, confirming that for a transport aircraft of this class, the requirements for stability, control, and structural integrity define less possible feasible designs, regardless of the number of design variables.

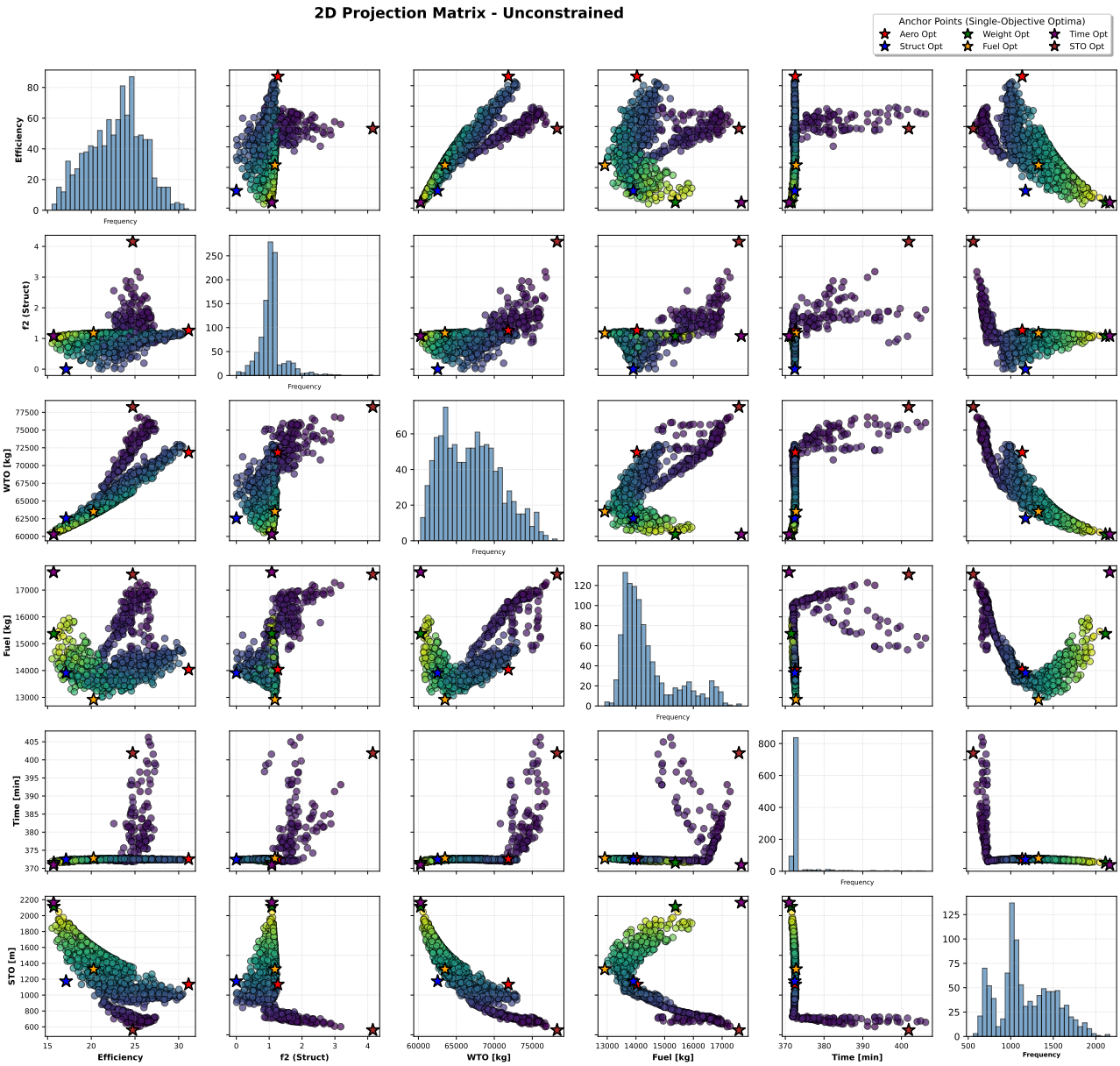


Figure 1.16: Scatter Plot Matrix of the Pareto Front (3 Design Variables) - **Unconstrained**.

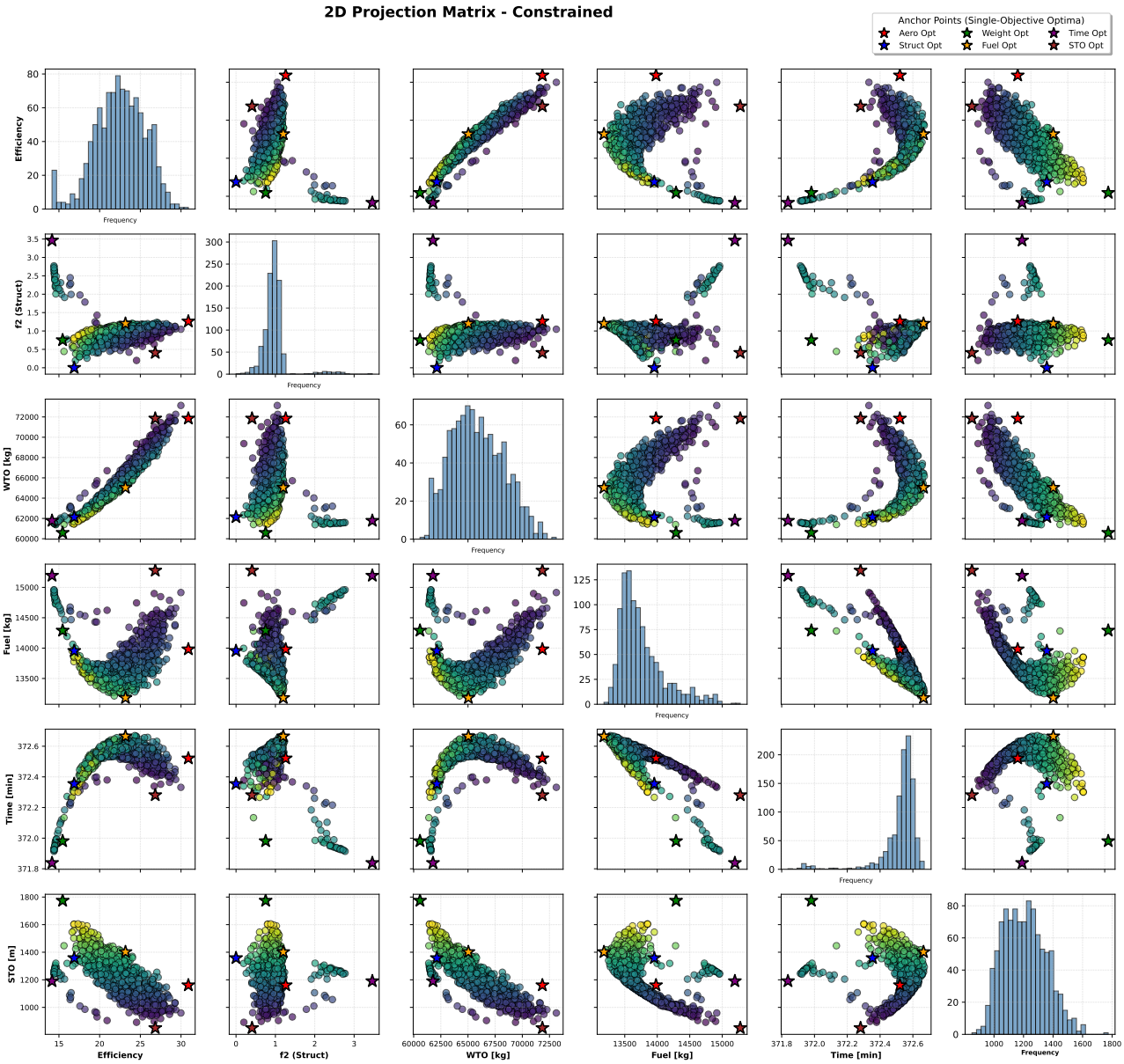


Figure 1.17: Scatter Plot Matrix of the Pareto Front (3 Design Variables) - **Constrained**.

1.3.6 Interactive Visualization Support Tool

To facilitate the visualization, an interactive software tool was developed. This application allows for real-time exploration of the Pareto front, enabling the designer to visualize how changing the relative importance of the objectives influences the optimal aircraft geometry and performance.

Pareto Front Interpolation

The Normal Boundary Intersection (NBI) method generated a discrete set of Pareto-optimal points. To allow for continuous exploration of the design space, an interpolation strategy was implemented to map the objective weight vector $\mathbf{w}$ directly to the optimal design variables $\mathbf{x}^*$.

Let $\mathcal{D}$ be the dataset of optimal solutions generated by the NBI algorithm, where each solution k is associated with a specific weight vector used during the optimization (or implied by its position on the front). A multivariate interpolation model $\mathcal{M}$ was trained on this dataset:

$$\mathbf{x}^* = \mathcal{M}(\mathbf{w}) \tag{1.13}$$

where $\mathbf{w} \in \mathbb{R}^6$ is the vector of objective preferences and $\mathbf{x}^*$ contains the optimal span (b), root chord (c_r), and optionally the wing position (X_{wing}). This approach decouples the visualization from the optimization process. Since the costly optimization is pre-calculated, the interpolation model provides instantaneous feedback ($\approx$ milliseconds), allowing for a smooth user experience. Two distinct interpolation sets were created: one for the fixed-wing configuration (2 variables) and one for the variable-position configuration (3 variables).

Software Architecture

he tool visualizes results from multi-objective optimizations with the following objectives:

Objective	Type	Description
Max Efficiency (E)	Maximize	Max Trimmed Aerodynamic Efficiency
STO	Minimize	Takeoff distance according to FAR25 regulations
MWTO	Minimize	Aircraft total estimated maximum take off weight
Fuel Consumption	Minimize	Total fuel required for mission
Mission Time	Minimize	Total mission duration

Constraint	Requirement	Purpose
SSM (Static Margin, sign-reversed)	> 0.15	Ensures longitudinal static stability
Horizontal tailplane volumetric coefficient (η_h)	$0.7 \leq \eta_h \leq 1.25$	Guarantees longitudinal stability and controllability

The optimization uses a weighted sum approach where **relative ratios** determine the importance of each objective, not absolute values.

- Setting weights [100, 0, 0, 0, 0] produces the **same result** as [2, 0, 0, 0, 0] (both prioritize only efficiency).
- Setting [1, 1, 0, 0, 0] means equal importance between efficiency and STO.

Figure 1.18 shows the interface of the developed tool. The user can switch between "Constrained" and "Unconstrained" optimization modes and enable or disable the wing position variable. To ensure portability and ease of distribution, the tool was compiled into a standalone executable and the source code was published on a dedicated GitHub repository.

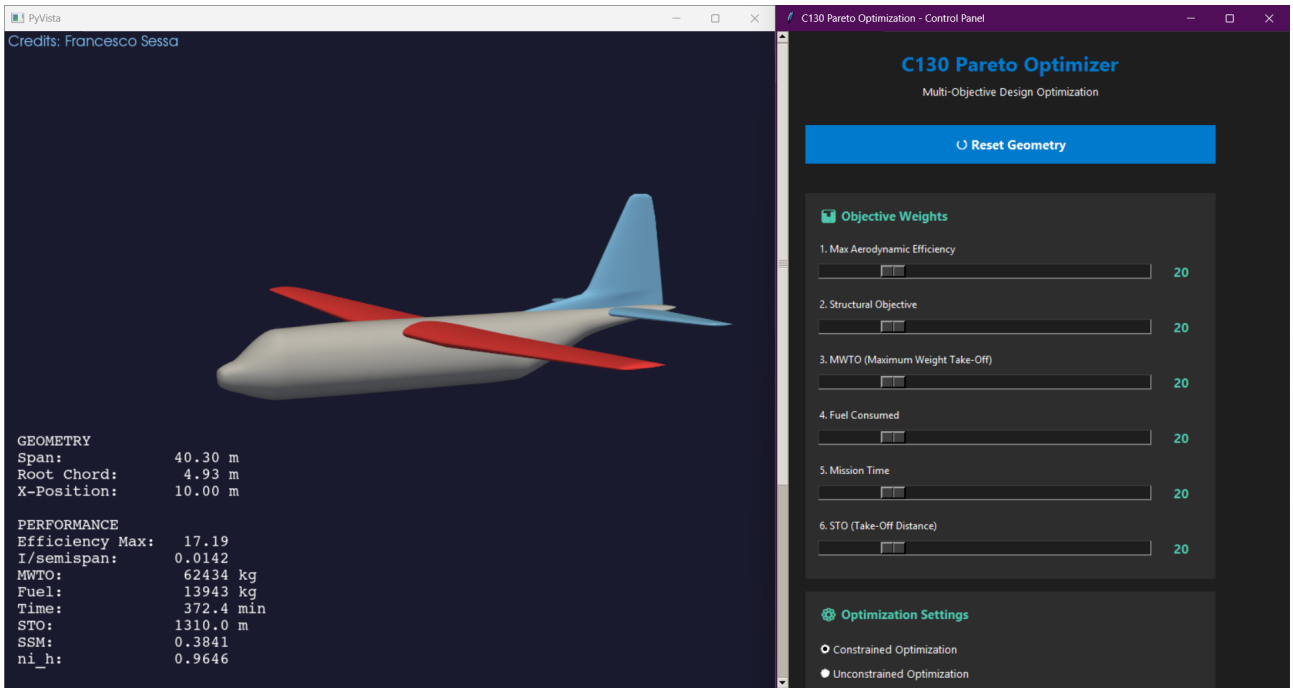


Figure 1.18: The Interactive Pareto Visualizer.