A Variable-Oriented Morphological Analysis to Generate Architectures for Improved Multidisciplinary Optimization Process

C. P. Frank[†], *O. J. Pinon-Fischer and D. N. Mavris*
Georgia Institute of Technology
270 Ferst Dr, Atlanta, Georgia, 30332-0150, United States
christopher.frank@gatech.edu

Abstract

The development of new aerospace vehicles requires the exploration of immense design spaces. To support such task, a systematic generation of concepts that can be efficiently compared and optimized is critical. This paper presents a variable-oriented morphological analysis which, when implemented with a functional decomposition, helps generate and group all feasible solutions into architectures. When applied to suborbital vehicles, the number of alternatives to be explored is reduced from 2.7 million to only 13 architectures, while enabling a higher number of configurations to be investigated. Finally, the execution time of the optimization process is reduced by a factor of 10^5.

1. Introduction and motivation

Recent technological developments have resulted in the emergence of new advanced vehicles such as flying cars, suborbital vehicles, and hypersonic aircraft. As discussed by Frank et al.,[1] these new markets are characterized by a significant diversity in existing concepts. The complexity of these new vehicles gives rise to a large combinatorial space of possible configurations for which no baseline has been established. In the last few years, various configurations of similar concepts have been developed. However, no design space exploration or global optimization has been performed. Instead, major design decisions were made based on expertise and company's experience and belief of what would work or not. Indeed, due to the novelty of such vehicles, companies logically favored the use of technologies they had already developed and leveraged their own internal skills and know-how to design their vehicle. Villeneuve[2] showed that this lack of rigorous methodology has also been the "norm" in the design of launchers that emerged in the last decades. There is thus a need for a rigorous and systematic methodology that enables the exploration of a large combinatorial design space and supports quantitative trade-off analyses to facilitate the selection of a design at a conceptual design level.

Three different design approaches have been identified in the literature: the typical aircraft design process,[3–7] the architecture optimization[2,8] approach and the architecture comparison[9–14] approach. Figure 1 illustrates the ability of these current design approaches to explore the design space. The available design space is represented in beige. The blue circles represent the areas of the design space covered by the corresponding design approach. The size of the circles is proportional to the number of configurations considered by a given design process. Consequently the larger the circle, the larger the number of design variables considered. The intensity of the color is a measure of the level of detail of the associated modeling technique. The typical aircraft design approach is built around a baseline and infuses new technologies to meet the stated requirements. Hence, it considers few configurations around a single baseline architecture but the corresponding analysis framework is extremely accurate and detailed. This leads to a single dark circle with a relatively large size. The architectures comparison approach compares many architectures but the level of detail of the evaluation framework is often based on qualitative information. In addition, only one configuration per architecture is considered and there is no optimization. This results in numerous small light blue circles. The architectures optimization approach considers fewer baseline architectures and performs separate optimizations before comparing the best configurations for each architecture. While numerous configurations are considered for each architecture, the modeling environment is more accurate than the "architectures comparison" approach but less accurate than the typical aircraft design approach. This leads to 2-3 large and relatively light blue circles.

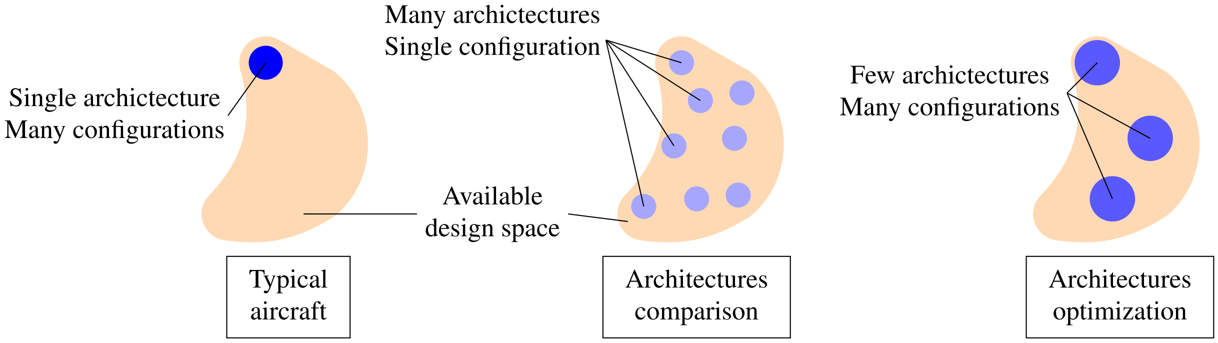


Figure 1 – Performance of current aerospace design processes[1]

While these approaches have their own advantages, they fail to cover the entire design space. As a consequence, there is a risk of missing promising concepts due to the lack of exhaustiveness during the alternative generation process. In addition, since revolutionary concepts are considered, one cannot rely on experience and experts' judgment to select a baseline. To address this limitation, a new methodology is needed that systematically generates all feasible alternatives in such a way that they can be further optimized and compared by a multidisciplinary optimization algorithm based on design variables specific to each architecture. A combination of the "architectures comparison" approach and the "architectures optimization" approach would provide the required capabilities. Indeed, such methodology would be able to consider and compare as many architectures as the "architectures comparison" methods by using their capability to systematically generate alternatives. This capability would translate into a large number of circles in the design space. The methodology would also be able to optimize configurations within each architecture by considering multiple variables for each subsystem similar to the "architectures optimization" methods. This capability corresponds to larger circles. Finally, in order for the impacts of each design concept to be captured, an accurate modeling and simulation environment must be implemented through physics-based modeling. This last capability is translated into darker circles. Figure 2 represents the desired performance of the new methodology in terms of design space exploration capabilities.

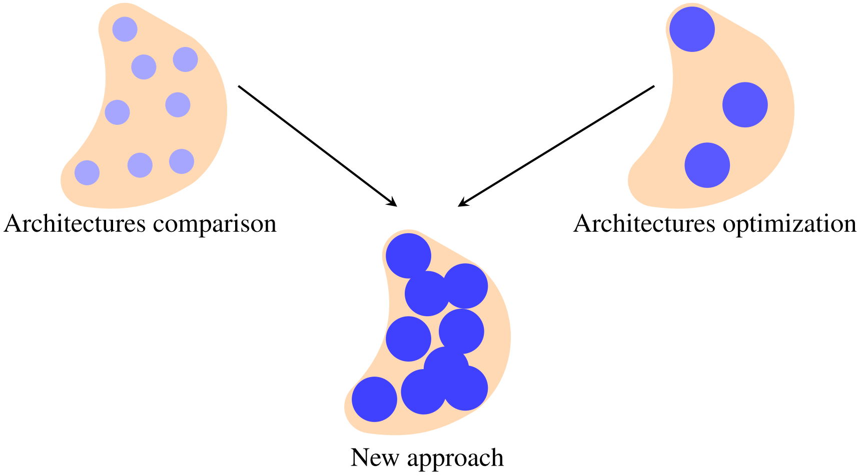


Figure 2 – Expected capabilities of the new approach

Three key challenges must be overcome to develop this new approach. First, in order to cover the entire design space, there is a need for a systematic and rigorous process that generates all feasible alternatives. The methodology needs to go beyond the simple brainstorming that only lists all possible alternatives that exist in the literature. Then, in order for alternatives to be optimized by a single algorithm, they need to be described by the same variables, which is not the case for most of complex aerospace vehicles. Finally, since optimization processes might be long to run, the number of algorithms to be executed has to be as small as possible. Frank et al.[1] proposed a new design space exploration methodology that aims at overcoming the aforementioned challenges. A brief overview of the methodology is presented in Figure 3. First, alternatives are generated following a systematic and rigorous process. Then, they are grouped into architectures, which correspond to groups of alternatives that are defined by the same variables. Each of these architectures is then optimized using a meta-heuristic multi-objective optimization algorithm. This results in multiple Pareto frontiers that are combined using an evolutionary algorithm. The latter drives each architecture optimization based on the architecture performance in order to only favor promising architectures and therefore improve the efficiency of the overall design process. The final output is a combination of the best alternatives from each architecture via an overall Pareto frontier.

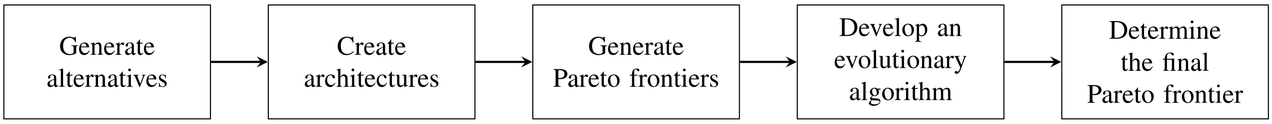


Figure 3 – Proposed approach for design space exploration

As mentioned, one of the key enablers of this methodology is the generation of architectures, defined in the context of this paper as "groups of alternatives described by the same design variables". Hence, the objective of this work is to support the aforementioned methodology by developing an efficient process that generates architectures by incorporating design variables for an improved multidisciplinary optimization process. To reach this objective, a review of existing alternatives generation methodologies is first conducted. Then, the proposed theoretical approach is described along with its implementation. Finally, suborbital vehicles are used as a proof-of-concept to demonstrate the capabilities and the benefits of this architecture generation process.

2. Review of previous work

Designers traditionally rely on their experience and best practices to generate alternatives and to identify the best solution to a given problem. However, this approach cannot be used when designing revolutionary concepts since no historical data are available. Such approach also leads to a very limited number of alternatives and does not favor innovation. In addition, choices tend to be biased by designers' experience and the process is extremely time consuming. Several methods have been developed in order to overcome these pitfalls. They are described, compared and evaluated in the following sections.

2.1 The 6-3-5 method

Created by Rohrbach[15,16] in 1968, this method aims at generating 108 new ideas in half an hour. It requires six designers who sit together with each participant having to write three ideas every five minutes. Ideas are then passed on to the next designer who uses it for inspiration to create more ideas. Hence, after six rounds, a total of 108 ideas have been generated. This method encourages quantity without ensuring quality. Moreover, there is no rigorous or systematic process involved and the ideas rely on experience of the designing team.

2.2 Design catalogues

This method involves selecting a handful of representative solutions instead of all possible ones and is very popular in scenarios development. It only provides a limited number of alternatives to consider for further analyses.[17] For example, instead of considering billions of combinations of price scenarios, only three are considered: baseline, optimistic and pessimistic. Using this method, designers will miss some opportunities and will not be able to screen the entire design space. In addition, the creation of these representative solutions requires experience in the field, which is not available when designing revolutionary concepts aimed at conquering new markets.

2.3 Theory of Inventive Problem Solving (TRIZ)

Created by Altshuller,[18] this methodology is built around five steps: identify the problem, formulate the problem by identifying potential gaps or technical difficulties, search for previously well-solved problems, identify analogous problems with known solutions, and adapt identified solutions to the stated problem. In doing so, Altshuller classified problems into five categories based on their degree of inventiveness. He also defined the percentage occurrence of each of them: apparent solution (32%), minor improvement (45%), major improvement (18%), new concept (4%) and discovery (1%). As a consequence, this methodology would work for about 95% of the problems. Finally, while widely implemented,[19–21] this methodology cannot be applied to innovative concepts and does not provide a way to systematically generate concepts.

2.4 Morphological matrix

The General Morphological Analysis (GMA), developed by Zwicky,[22–24] is a methodology that aims at exploring all possible solutions in multi-dimensional and non-quantifiable complex problems. One of its most common form is the morphological matrix used in numerous aerospace design methodologies[4,7,25,26]: the system is decomposed into functions or features (lines in the morphological matrix) for which different alternatives are identified (column in the morphological matrix). Then, solutions are created by generating all combinations of alternatives. This full-factorial approach to design space definition is rigorous and systematic but also generates non-feasible combinations. Hence, this method is usually combined with a compatibility matrix. The latter contains only ones and zeros in order to mention if two alternatives are compatible (1) or not (0). Even though combining a morphological matrix with a compatibility matrix helps remove infeasible solutions, the number of feasible alternatives generated is usually extremely large.

2.5 Interactive Reconfigurable Matrix of Alternatives (IRMA)

The aforementioned morphological matrix has been improved to account for the dynamic and iterative nature of decision-making processes.[27–30] Indeed, the IRMA embodies both the morphological and the cross-consistency matrices within a single software framework. It benefits from the following advantages: hidden integrated compatibility matrices, filters that can reduce the number of alternatives, standardized and flexible, integrated method for alternatives selection, etc. However, it does not track design variables and the number of possible solutions generated is still extremely large.

2.6 Adaptive Reconfigurable Matrix of Alternatives (ARM)

Even though the IRMA greatly supports the generation of design concepts, it is based on a static functional decomposition. To overcome this pitfall, ARM[9] has been developed that relies on functional induction. It acts as a hybrid of the IRMA and the function/means tree so that both functional and physical breakdowns can be interactively managed. While being more flexible than the IRMA, it suffers from the same main pitfall: design variables are not tracked. In addition, this tool aims at supporting the designers in the concept selection rather than in the concept generation. The ARM has been implemented in the Architecture Design Environment[9] (ADEN), whose goals were to manage complex relationships between architectural elements and to provide a framework for trade-off analyses and performance evaluation of a single generated alternative.

2.7 Decision tree

Decision trees map out all possible paths that can be followed to achieve a primary goal and its corresponding subgoals.[31] Hence, if goals are linked to vehicle features, decision trees can help lay out alternatives. This approach is systematic and logic and thus reduces the probability to omit items. Compatibility issues are directly addressed so that only feasible solutions are generated at the end nodes. This approach is similar to the morphological matrix except that it requires designers to individually generate all alternatives. Hence, even if it does not require a separate compatibility matrix, the same amount of thinking is needed from the designers. However, it might lead to more missing alternatives than the morphological matrix and does not track design variables.

2.8 Morphological Evaluation Machine and Interactive Conceptualizer (MEMIC)

MEMIC[32] is an automated concept generator able to produce multiple design solutions from a given set of subfunctions. It relies on a web-based repository. The tool uses function-component relationships embedded in a function-component matrix and the component-component compatibility contained in a Design Structure Matrix (DSM). This tool aims at assisting designers in the choice of the various options for each function and sub-function while also supporting the generation of several “good” solutions. However, it does not include a systematic generation of alternatives in order to cover the entire design space and does not capture the design variables that define the different alternatives.

2.9 Method selection

This review indicates that the best way to explore as many alternatives as possible is to decompose the system into features. In particular, the combination of both morphological and compatibility matrices seems to provide the best approach to a systematic and rigorous generation of alternatives. However, the resulting number of alternatives that must be considered for further optimization and comparison remains extremely high. Since the multi-objective optimization of each concept requires a significant amount of setup and execution time, it becomes rapidly impractical to use this process alone. In addition, this process does not consider design variables for a downstream use of the generated alternatives. Those two challenges are addressed in the following section through the development of an improved morphological matrix.

3. Approach

The objective of the new method is to generate all feasible alternatives in a way that they can be efficiently optimized. While the typical morphological matrix provides a good method to generate all possible alternatives, it does not ensure the feasibility of the obtained alternatives. To overcome this drawback, the morphological matrix is combined with a compatibility matrix. The combination is able to provide all feasible alternatives but does not account for design variables. Since a single optimizer can only handle alternatives that are defined by the same design variables, there is a need for grouping them. Hence, alternatives that are defined by the same design variables are grouped into a single architecture. For instance, straight wings, delta wings and swept wings are grouped into a single wing category and are defined by their sweep angle, surface area, aspect ratio, etc. This allows for each generated architecture to be optimized by a single optimizer. The output of this method is a list of all feasible architectures and their corresponding design variables. The development of the aforementioned method can be decomposed into four steps:

1. Generate alternatives: alternatives are generated using a morphological analysis. This method is particularly useful for multi-dimensional complex problems since it provides a structured, functional and intelligent way to decompose a problem and generate alternatives[33]. One of the most common practices in morphological analysis is the use of a morphological matrix, or matrix of alternatives. This matrix is a two-dimensional representation of the system. Each row represents a function/feature of the system and each column represents an option. The number of possible alternatives is given by the product of the number of options of all functions. If the morphological matrix M is defined by its rows i and its columns j, the number of alternatives is defined in Eq. 1.

$$\mathrm{N_{alt}} = \prod_i \left(\sum_j M^*_{i,j} \right) \quad \text{where} \begin{cases} M^*_{i,j} = 1 & \text{if } M_{i,j} \neq \emptyset \\ M^*_{i,j} = 0 & \text{if } M_{i,j} = \emptyset \end{cases} \tag{1}$$

2. Ensure feasibility: feasibility is ensured by only selecting the combinations of options for which all options are compatible with each other. A square matrix called compatibility matrix is used to define compatibility between options. This matrix C is symmetric so that only the upper (or lower) triangular needs to be completed by either 0 or 1 according to the rule presented in Eq. 2. In this equation, k and l represent the k^{th} row and the l^{th} column of the matrix.

$$C_{k,l} = \begin{cases} 1 & \text{if options } k \text{ and } l \text{ are compatible} \\ 0 & \text{if options } k \text{ and } l \text{ are incompatible} \end{cases} \tag{2}$$

   Using the definition of the compatibility matrix, the diagonal can be prefilled by ones since a given option is necessarily compatible with itself. In addition, while functions can be described by different options, only one option per function can be selected for a given design alternative. As a consequence, all options used to define the same function are necessarily incompatible. Therefore, additional entries can be prefilled. As a consequence, the final number of entries that must be filled by the designer is reduced to N_f, as defined in Eq. 3.

$$N_f = \frac{N_{alt}\,(N_{alt} - 1)}{2} - \sum_i \frac{\left(\sum_j M^*_{i,j}\right)\left(\left(\sum_j M^*_{i,j}\right) - 1\right)}{2} \tag{3}$$

3. Assign variables: the design variables need to be identified to enable alternatives to be grouped into architectures. First, all design variables that are used by at least one option of a specific function are defined. Then, the listed variables are assigned to each option using assignment matrices. These matrices are composed of checkboxes to link variables to options, as displayed in Figure 5.

4. Generate feasible architectures: based on both the morphological matrix and the compatibility matrix, an algorithm will first determine the list of feasible alternatives. Then, based on this list and the assigned design variables, alternatives that are described by the same variables are grouped into architectures.

This methodology has been implemented into a new software written in Matlab called EfficieNt Variable-orIented Software for Architecture GEneration (ENVISAGE). Its development is described in the following section.

4. Development of ENVISAGE

The general structure of the environment, described in Figure 4, is based on the method discussed in the previous section. First, the users input their morphological matrix (Table 1) based on a functional decomposition.

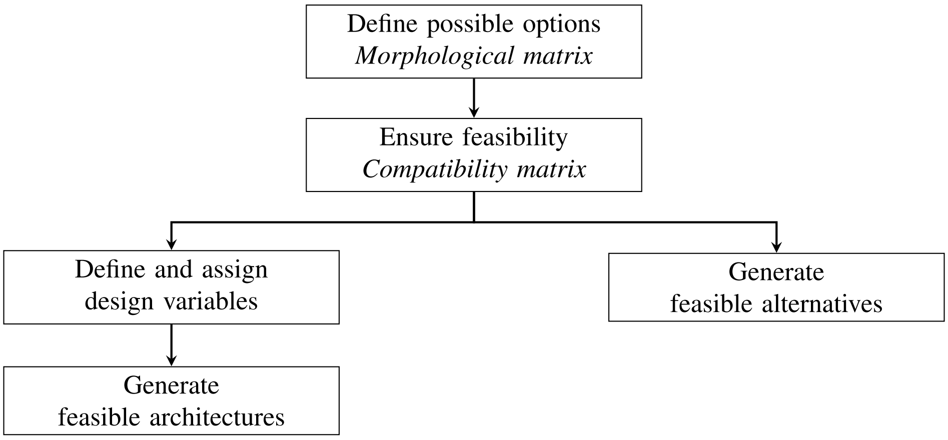


Figure 4 – General structure of ENVISAGE

The first window allows users to build their morphological matrix. A notional 3×3 matrix is presented in Table 1 with notional feature names. The users can add both features and options and change their names using the corresponding buttons.

Table 1 – Morphological matrix

	Option 1	**Option 2**	**Option 3**
Feature 1	Option A	Option B	
Feature 2	Option C	Option D	Option E
Feature 3	Option F	Option G	

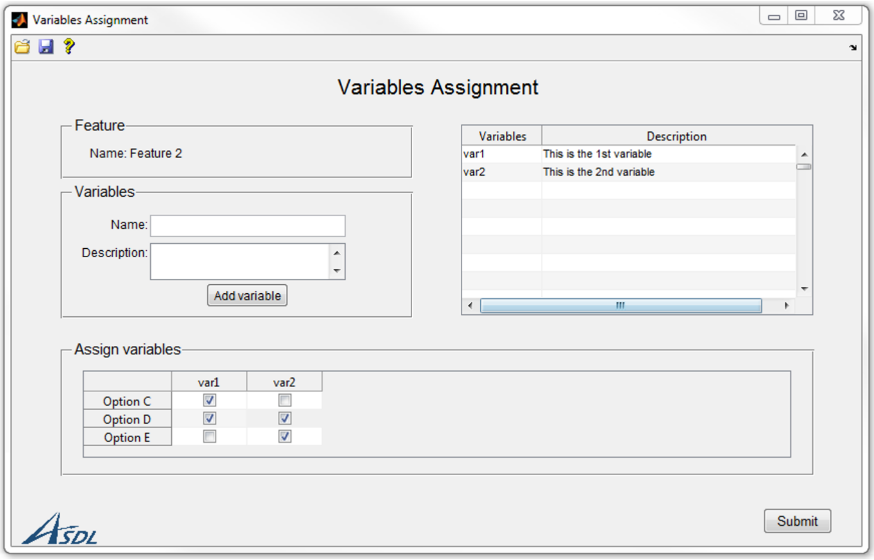
Then, to ensure the feasibility of the generated concept, they complete a pre-filled compatibility matrix. Table 2 displays the prefilled matrix for a notional problem. ENVISAGE also allows users to upload and save each matrix in Microsoft Excel®.

Table 2 – Prefilled compatibility matrix

	Option A	**Option B**	**Option C**	**Option D**	**Option E**	**Option F**	**Option G**
Option A	1	0					
Option B		1	0				
Option C			1	0	0		
Option D				1	0		
Option E					1		
Option F						1	0
Option G							1

Once the compatibility matrix has been filled, two analysis options are proposed:

- An alternative-based analysis: users can generate the list of all feasible alternatives
- An architecture-based analysis: after defining the design variables specific to each function and option (Figure 5), the users can generate the list of all feasible architectures. The users can also add variables along with a description through the panel “Variables”. The variables are then added to the table on the left-hand side, allowing the users to attribute them to the corresponding options through check-boxes. The toolbar also allows users to save and load the variables/description table.



Variables Assignment

Feature

Name: Feature 2

Variables

Name:

Description:

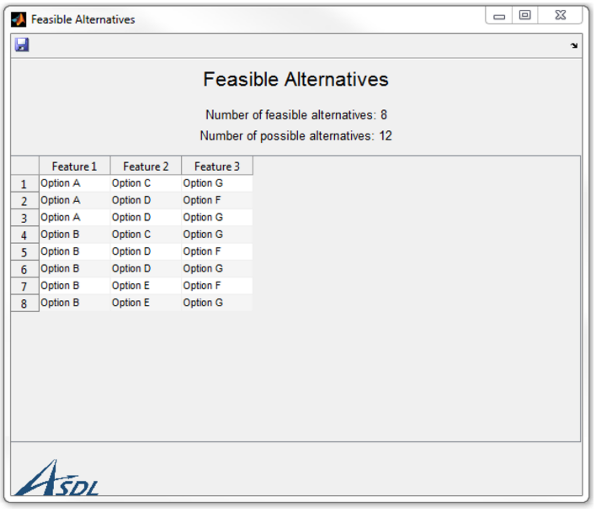
Variables	Description
var1	This is the 1st variable
var2	This is the 2nd variable

Assign variables

	var1	var2
Option C	☑	☐
Option D	☑	☑
Option E	☐	☑

Figure 5 – Variable assignment window of ENVISAGE

The list of feasible alternatives (Figure 6) is generated by a recursive function that searches for feasible combinations of options. For each visited option, the recursive function checks the compatibility between the visited option and all other options already in the list. If the users want architectures to be formed, a function re-orders the aforementioned list using a binary representation of the design variables in order to group alternatives that are defined by the same variables (Figure 7). In both cases, users can save the final table through the toolbar and visualize the list of all variables along with their description.



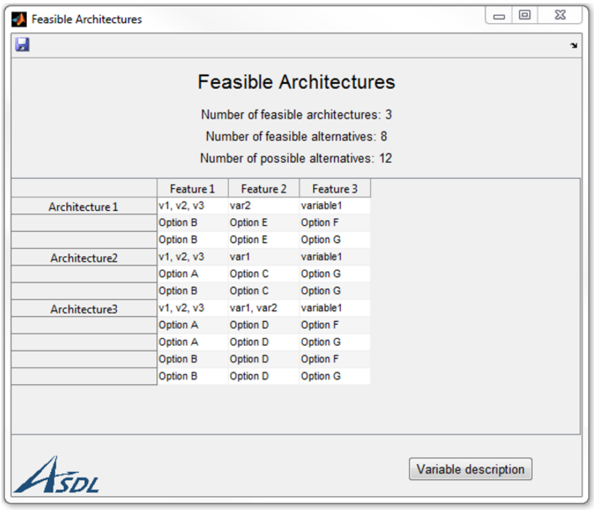
Feasible Alternatives

Number of feasible alternatives: 8

Number of possible alternatives: 12

	Feature 1	Feature 2	Feature 3
1	Option A	Option C	Option G
2	Option A	Option D	Option F
3	Option A	Option D	Option G
4	Option B	Option C	Option G
5	Option B	Option D	Option F
6	Option B	Option D	Option G
7	Option B	Option E	Option F
8	Option B	Option E	Option G

Figure 6 – List of feasible alternatives



Feasible Architectures

Number of feasible architectures: 3

Number of feasible alternatives: 8

Number of possible alternatives: 12

	Feature 1	Feature 2	Feature 3
Architecture 1	v1, v2, v3	var2	variable1
	Option B	Option E	Option F
	Option B	Option E	Option G
Architecture2	v1, v2, v3	var1	variable1
	Option A	Option C	Option G
	Option B	Option C	Option G
Architecture3	v1, v2, v3	var1, var2	variable1
	Option A	Option D	Option F
	Option A	Option D	Option G
	Option B	Option D	Option F
	Option B	Option D	Option G

Figure 7 – List of feasible architectures

The following section discusses the implementation and benefits of using a variable-oriented morphological matrix in the design of manned suborbital vehicles.

5. Application to the design of suborbital vehicles

The main objective of commercial suborbital vehicles is to carry passengers to an altitude higher than 100 km. To reach the targeted altitude, the vehicle must take-off (or be launched), fly and land safely. To be able to fly, the vehicle can create lift and/or thrust. A rocket engine is needed to ensure that the vehicle reaches a high velocity and high altitude. The rocket engine can, in turn, be seconded by different types of jet engines. In addition, the vehicle must also be laterally and longitudinally stable and controllable in the atmosphere and in space. This functional decomposition is illustrated in Figure 8. The gray boxes correspond to the end functions that will appear in the morphological matrix, while the white boxes correspond to high-level features that are further decomposed.

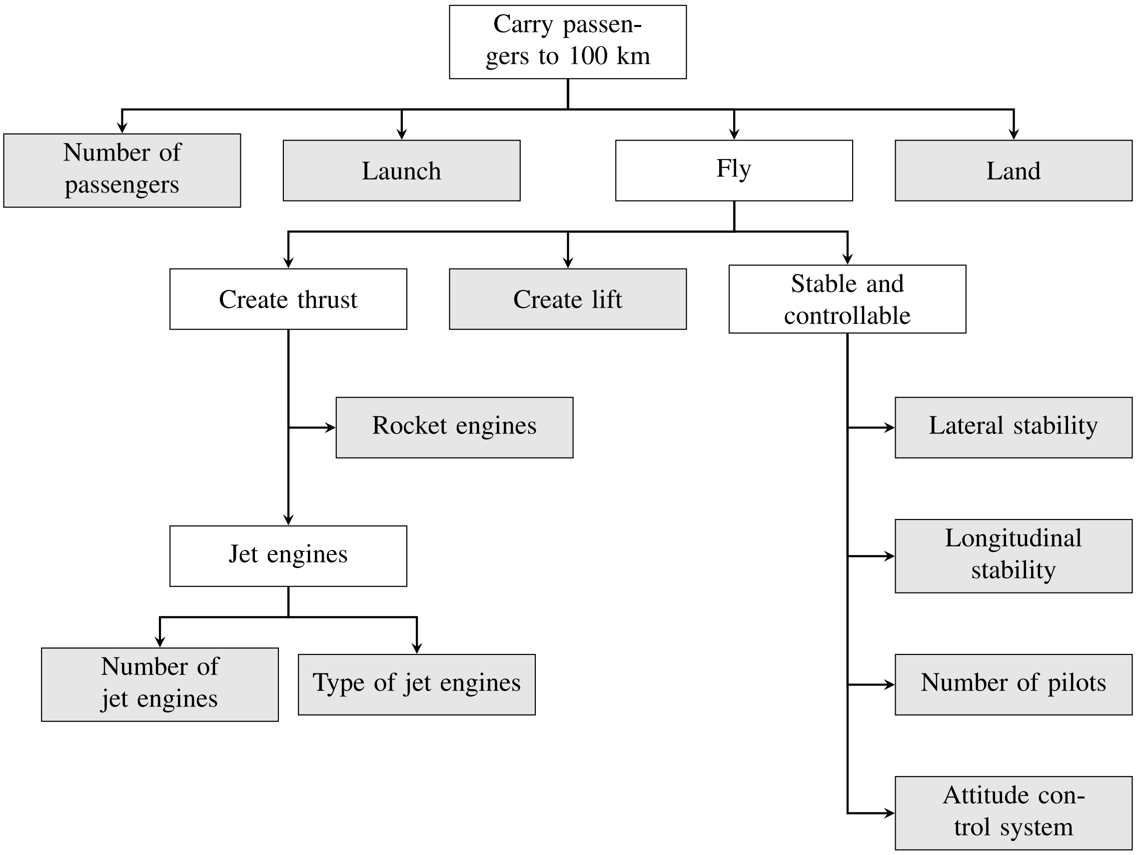


Figure 8 – Functional decomposition of suborbital vehicles

Using the previously defined functions, the options can be generated by looking at existing concepts. As such, more than 30 vehicles have been investigated and decomposed in order to find the possible options.[34–37] The following description provides the required data to build the morphological matrix for suborbital vehicles (Table 3), such as the ones illustrated in Figure. 9:

(a) Airbus D&S Space Jet[38] (b) SpaceShipOne[39] (c) Black Armadillo[40]

Figure 9 – Existing concepts of suborbital vehicles

- Type of launch: suborbital vehicles can take-off horizontally (similar to aircraft) or vertically (similar to rockets). In addition, some concepts have been launched by an intermediate vehicle: either an aircraft (SpaceShipOne) or a balloon (Wild Fire).

- Type of landing: suborbital vehicles can land horizontally with power (similar to aircraft) or without power (similar to the SpaceShipOne and the Space Shuttle). In addition, some concepts can also land using retro-rockets (New Shepard) or a parachute (Black Armadillo) to slow down their rate of descent.

- Lift generation: both slender bodies and winged bodies have been used in existing concepts. Slender bodies do not generate lift and only rely on their thrust to fly such as Black Armadillo. Winged bodies can be equipped by straight wings (SpaceJet), delta wings (Vehra) or swept wings (Rocketplane XP).

- Longitudinal stability: for vehicles equipped with wings, there might be a need for a second horizontal lifting surface that helps controlling the stability of the vehicle. This can be fulfilled either by an horizontal stabilizer or by a canard.

- Lateral stability: for vehicles equipped with wings, there is a need for a vertical surface that controls the lateral stability of the vehicle. This can be fulfilled either by a vertical stabilizer or by large wing tips.

- Type of rocket engines: the three main types of chemical rocket engines (solid, liquid, and hybrid) can be used to power a suborbital vehicle from low altitude to around 100 km. Also, because the range of thrust used in suborbital vehicles corresponds to the trade-off zone between pressurized and pump-fed liquid engines, both types must be considered in addition to solid and hybrid rocket engines.

- Number of jet engines: in order to benefit from an efficient propulsion at low altitude, the rocket engine can be seconded by jet engines. A maximum of 4 jet engines can be considered, similar to the biggest commercial aircraft in service.

- Type of jet engines: since the jet engine has to operate for a short period of time and provide a large amount of thrust, to allow for the vehicle to fly at high speeds and high altitude, only turbojet and turbofan engines are considered.

- Number of pilots: while most of the concepts include human pilots (1 or 2), some of them are fully automated such as the New Shepard.

- Number of passengers: While all commercial suborbital vehicles aim at carrying passengers (or at least an equivalent weight of payload), the number greatly varies between concepts. Indeed, it goes from 1 (Ascender) to 8 (Space Cruise).

- Attitude control system: in order to prepare the re-entry to be controllable at high altitude, the vehicle must be equipped with an attitude control system. According to existing concepts, the latter can be either composed of cold gas engines (Canadian Arrow) or liquid rocket engines (Lynx 2).

Based on this literature review, the morphological matrix presented in Table 3 is inputted in the software. A first raw alternatives generation would result in a total of 2,764,800 possible combinations. However, among these combinations some are incompatible. Hence, the next step aims at removing all incompatible combinations from the generated list.

The compatibility between options is set based on the authors' knowledge and judgment. For example, a slender body cannot take-off or land horizontally. In addition, there is no incentive to carry jet engines if the vehicle is launched from an aircraft or from a balloon. Other similar considerations are made to build the compatibility matrix. Once completed, ENVISAGE is executed in order to extract all feasible alternatives. A list of 123,000 feasible alternatives is provided.

Table 3 – Morphological matrix for suborbital vehicles

	Option 1	**Option 2**	**Option 3**	**Option 4**	**Option 5**	**Option 6**	**Option 7**	**Option 8**	**Number of alt.**
Type of launch	Horizontal	Vertical	Aircraft launched	Balloon launched					4
Type of landing	Horizontal powered	Gliding	Rocket	Parachute					4
Lift generation	Delta	Swept wing	Straight wing	None					4
Longitudinal stability	Horizontal stabilizer	Canards	None						3
Lateral stability	Vertical stabilizer	Wing tip	None						3
Type of rocket engines	Pressurized liquid	Pump-fed liquid	Solid	Hybrid					4
Number of jet engines	0	1	2	3	4				5
Type of jet engines	Typical turbojet	Augmented turbojet	Typical turbofan	Augmented turbofan	None				5
Number of pilots	0	1	2						3
Number of passengers	1	2	3	4	5	6	7	8	8
Attitude control system	Cold gas	Liquid							2
Number of possible combinations									**2,764,800**

5.1 Variables definition and assignment

To create architectures, the design variables used to describe each function have to be listed and assigned to their specific options. The list of all design variables considered in this study is provided below:

- Type of launch: initial speed V_i, initial altitude h_i, take-off field length L_{TO}
- Type of landing: landing field length L_{LA}, landing speed V_{LA}, engine re-start altitude t_r
- Lift generation: reference area S_{ref}, sweep angle Λ, aspect ratio AR
- Longitudinal stability: horizontal surface area S_h, maximum thickness-to-chord ratio of the horizontal surface $t_{c,h}$
- Lateral stability: vertical surface area S_v, maximum thickness-to-chord ratio of the vertical surface $t_{c,v}$
- Type of rocket engines: propellant $prop$, nozzle expansion ratio ϵ, chamber pressure p_c, thrust T_r, burning time t_b
- Number of jet engines: number of jet engine n_j
- Type of jet engines: afterburner ab, turbine inlet temperature T_4, thrust T_j, transition altitude h_t
- Number of pilots: number of pilots n_p
- Number of passengers: number of passengers n_{PAX}
- Attitude control system: thrust T_{AC}, number of engines n_{AC} and burning time t_{AC}

Once the design variables are listed and assigned to the corresponding options, ENVISAGE is executed and identifies 13 feasible architectures. A summary of the architecture generation process is provided in Figure 10. Starting with about 2.8 million total combinations, the integration of the compatibility matrix within the process allows designers to divide the number of alternatives to be investigated by a factor of 23. Grouping all the alternatives defined by the same design variables into architectures further enables to reduce the number of architectures to be optimized to 13. Doing so allows for the number of discrete algorithms to be set to be reduced by a factor 215,000.

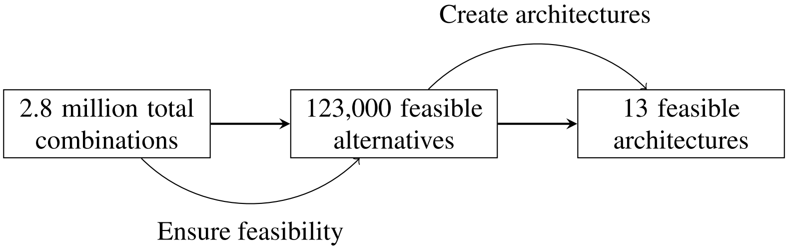


Figure 10 – Summary of the architecture generation process

The next section discusses and quantifies in more detail the benefits of implementing a variable-oriented morphological analysis.

6. Benefits of the new architecture generation methodology

As discussed in Section 1, a design space exploration usually requires complex optimization processes. The efficiency of such exploration is driven by the ability of the designers to both cover the maximum number of concepts and reduce the execution time to reach the best concept(s). The use of morphological analysis already ensures the exhaustiveness of the concept generation process. The proposed approach focuses on reducing the execution time of the optimization processes. The required number of function calls to reach the optimum value is used to better quantify the benefits of implementing a variable-oriented morphological analysis. Chelouah et al.[41] calculated the average number of function calls required for a genetic algorithm to optimize different test functions. Among these test functions, the Rosenbrock function R_n (Eq. 4) and the Zakharov function Z_n (Eq. 5) have n design variables and can be used to analyze the behavior of the optimization algorithm with respect to n. Since they can easily be scaled to n dimensions, these two functions provide a good way to measure the impact of increasing the number of design variables on the required number of function calls. In addition, they have been tested with the same convergence criteria so that the results are consistent.

$$R_n(x) = \sum_{j=1}^{n-1} \left[100 \left(x_j^2 - x_{j+1} \right)^2 + \left(x_j - 1 \right)^2 \right] \tag{4}$$

$$Z_n(x) = \left(\sum_{j=1}^{n} x_j^2 \right) + \left(\sum_{j=1}^{n} 0.5 j x_j \right)^2 + \left(\sum_{j=1}^{n} 0.5 j x_j \right)^4 \tag{5}$$

Their results for different values of n are presented in Table 4. Since the behavior of the problem's specific objective function is unknown, the average number of function calls is calculated for each n. It is then approximated by $f_c(n) = 7.7018n^2 + 1189.5n - 1323.6$ with an R^2 of 0.9995. Even though this relationship highly depends on the problem's objective function and constraints, it is assumed that the trend in the number of function calls with respect to the number of variables remains similar to the aforementioned functions.

Table 4 – Number of function calls for R_n and Z_n functions[41]

Functions	**Number of design variables**				
	2	**5**	**10**	**50**	**100**
R_n	960	3,990	21,563	78,356	194,302
Z_n	620	1,350	6,991	75,201	195,246
Average	790	2,670	14,277	76,779	194,774

The number of function calls required can now be compared assuming two design processes that can cover the entire design space: the first one optimizes all 2.8 million alternatives and the second one optimizes all 13 architectures generated by ENVISAGE. Figure 11 represents the computational time required to find the optimum value with respect to the number of design variables considered in the conceptual design of suborbital vehicles. It assumes an execution time of the design framework of one second. Using the traditional design process, the number of function calls N_t is defined by $N_t = 2.8 \times 10^6 \times f_c(n)$. Using ENVISAGE, $N_t = 13 \times f_c(n)$. As illustrated in Figure 11, implementing the aforementioned variable-oriented morphological analysis helps significantly reduce the required execution time, hence supporting the complete exploration of large design spaces.

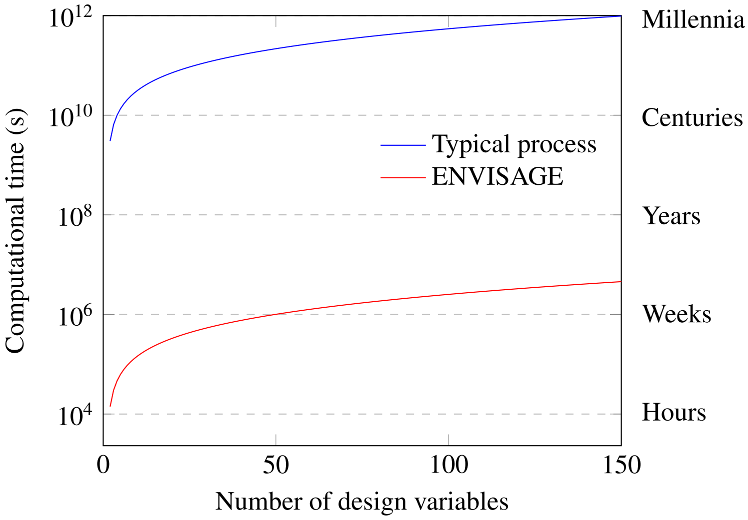


Figure 11 – Improvements in computational time

In order to estimate the benefit of grouping options against the disadvantage of adding more design variables, the factor k is introduced. k represents the number of variables that must be added to the optimization process when grouping two options. For instance, if $k = 5$, removing one option from a function requires 5 more variables in the optimization process. To perform trade-off analyses, a baseline morphological matrix is used, which is composed of 15 functions and 8 options per function, initially defined by 10 variables. Hence, the total number of combinations is $8^{15} = 3.5 \times 10^{13}$, which initially requires $8^{15} f_c(150) = 1.2 \times 10^{19}$ function calls to be completely investigated. Figure 12 displays the improvement in the number of function calls with respect to the number of options removed per function.

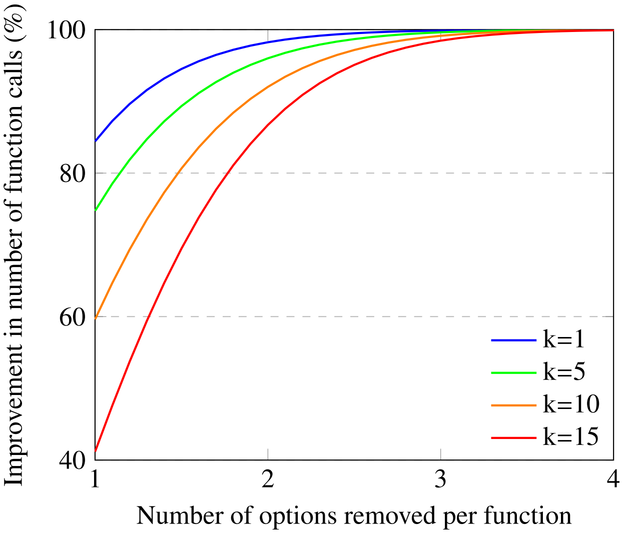


Figure 12 – Improvements in the number of function calls

As expected, the benefits are smaller as the number of variables added per function removed increases. The results also demonstrate that even for 15 variables added per option removed, the improvement exceeds 40%. In addition, grouping 3 out of the initial 8 options results in large improvements, independently of the number of variables added.

A sensitivity analysis is conducted to assess the sensitivity of improvement to changes in the number of functions, number of variables per function, etc. A Design of Experiments (DoE) is generated using the statistical software JMP®. A Latin Hypercube is used to cover the entire design space through 3,000 points. The variables and their corresponding ranges are presented below:

- Number of functions: [10; 30]
- Number of options per function: [5; 15]
- Number of variables per function: [2; 10]
- Number of variables added per option removed: [1; 15]
- Number of functions removed per function: [1; 5]

The results are presented in Figure 13. Each curve represents the sensitivity of the overall improvement in number of function calls with respect to changes in a specific parameter. For a given curve, this analysis assumes that all other parameters are constant. Therefore, each curve corresponds to partial derivatives of the improvement in number of function calls with respect to a specific parameter.

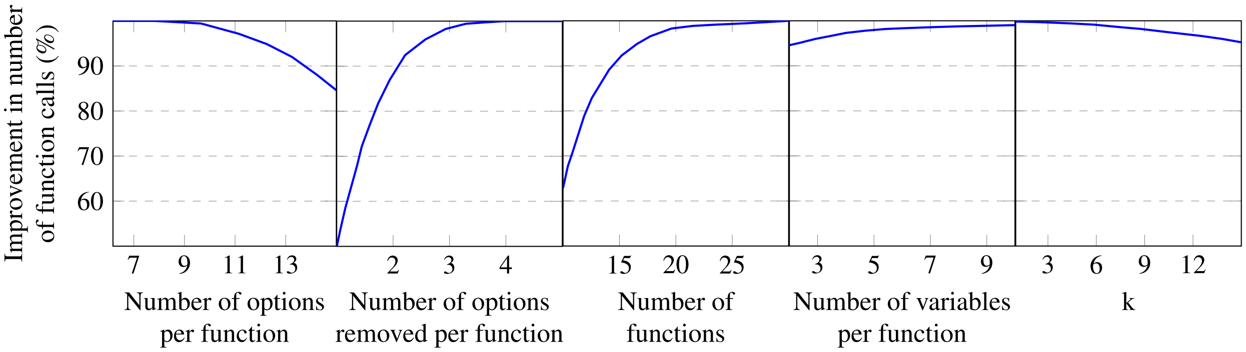


Figure 13 – Sensitivity of the overall improvement with respect to each parameter

As shown in Figure 13, the amplitude of the improvement in the number of function calls is mainly driven by the number of options removed per function and the number of functions considered in the morphological matrix. This analysis demonstrates that the improvements in the number of functions calls are more significant when the number of functions considered increases. In particular, it shows that the benefits of using a variable-oriented morphological analysis are more significant when dealing with larger problems that have a more detailed decomposition along with a higher number of design variables. These attributes characterize the design of complex concepts such as suborbital vehicles. The benefits of such morphological analysis decrease with the number of options per function and k, however their sensitivities are relatively small compared to the two key drivers.

7. Conclusion

While new aerospace vehicles open up immense design spaces, efficient design processes are required in order to ensure that the optimum vehicle has been selected. Frank et al.[1] propose a new methodology that relies on a parallel optimization and comparison of architectures. The architectures are broken down into groups of alternatives that are defined by similar design variables and can be optimized using a single optimization algorithm. This paper aimed at supporting this methodology by developing a systematic and rigorous approach to generate feasible architectures. This variable-oriented morphological matrix approach relies on both a morphological matrix and a compatibility matrix to generate all feasible alternatives. In particular, it groups all alternatives into architectures by infusing design variables within the process and assigning them to the corresponding options. This process has been implemented into a Matlab-based software called EfficieNt Variable-orIented Software for Architecture GEneration (ENVISAGE). The design of affordable manned suborbital vehicles for space tourism has then been used as a proof-of-concept for this approach. Around 2.8 million possible combinations have been identified through 45 options distributed into 11 options. The feasibility check enabled the removal of 95% of the possible combinations. Then, the grouping into architectures led to only 13 different architectures. Finally, the benefits of the proposed variable-oriented morphological analysis were quantified by the number of function calls required to reach the optimized concept. For the proposed problem, the number of function calls has been divided by a factor of 10^5. Finally, a sensitivity analysis showed that the benefits of the proposed approach were even more significant for larger problems that have a more detailed decomposition.

References

[1] C. P. Frank, O. J. Pinon-Fischer, and D. N. Mavris, "A design space exploration methodology to support decisions under evolving requirements' uncertainty and its application to suborbital vehicles," in *53rd AIAA Aerospace Sciences Meeting*, 2015.

[2] F. Villeneuve and D. N. Mavris, "A new method of architecture selection for launch vehicles," in *AIAA 13th International Space Planes and Hypersonics Systems and Technologies*, 2005.

[3] M. R. Kirby, *A Methodology for Technology Identification, Evaluation, and Selection in Conceptual and Preliminary Aircraft Design*. PhD thesis, School of Aerospace Engineering, Georgia Institute of Technology, 2001.

[4] D. N. Mavris and M. R. Kirby, "Technology identification, evaluation, and selection for commercial transport aircraft," in *Society of Allied Weight Engineers*, 1999.

[5] D. N. Mavris, D. S. Soban, and M. C. Largent, "An application of a technology impact forecasting (tif) method to an uninhabited combat aerial vehicle," in *4th World Aviation Congress and Exposition*, 1999.

[6] D. N. Mavris, "Technical feasibility and economic viability gap analysis." University Lecture, 2012.

[7] D. S. Soban and D. N. Mavris, "Assessing the impact of technology on aircraft systems using technology impact forecasting," *Journal of Aircraft*, vol. 50, no. 5, pp. 1380 – 1393, 2013.

[8] F. Villeneuve, *A Method for Concept and Technology Exploration of Aerospace Architectures*. PhD thesis, Georgia Institute of Technology, 2007.

[9] M. Armstrong, C. de Tenorio, E. Garcia, and D. N. Mavris, "Function based architecture design space definition and exploration," in *26th Congress of International Council of the Aeronautical Sciences (ICAS)*, 2008.

[10] B. Donahue, "Architecture selection: The key decision for human mars mission planning," in *37th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, 2001.

[11] F. A. Fossati and A. Denaro, "A rlv concept selection from the thermo-mechanical outlook the lesson learned in festip," in *9th International Space Planes and Hypersonic Systems and Technologies Conference and 3rd Weakly Ionized Gases Workshop*, 1999.

[12] A. P. Kothari and D. Webber, "Potential demand for orbital space tourism opportunities made available via reusable rocket and hypersonic architectures," in *AIAA SPACE 2010 Conference & Exposition*, 2010.

[13] N. Prasadh, R. Moss, K. Collett, A. Nelessen, S. Edwards, and D. N. Mavris, "A systematic method for sme-driven space system architecture down-selection," in *AIAA SPACE 2014 Conference and Exposition*, 2014.

[14] M. A. Walton and D. E. Hastings, "Applications of uncertainty analysis to architecture selection of satellite systems," *Journal of Spacecraft and Rockets*, vol. 41, no. 1, pp. 75 – 84, 2004.

[15] J. S. Linsey and B. Becker, *Design Creativity 2010*. Springer London, 2011.

[16] K. Otto and K. Wood, *Product Design*. Prentice Hall, Upper Saddle River, 2001.

[17] K. Roth, *Design catalogues and their usage*. Springer London, 2002.

[18] K. Gadd, *TRIZ for Engineers - Enabling Inventive Problem Solving*. John Wiley and Sons, Inc., 2011.

[19] M. Nordlund, "Applications of system theories and ai tools in aircraft design," in *5th Symposium on Multidisciplinary Analysis and Optimization*, 1994.

[20] M. Mayda and H. R. Borklu, "An integration of triz and the systematic approach of pahl and beitz for innovative conceptual design process," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 36, no. 4, pp. 859 – 870, 2013.

[21] J. Malmqvist, R. Axelsson, and M. Johansson, “A comparative analysis of the theory of inventive problem solving and the systematic approach of pahl and beitz,” in *Proceedings of The 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, 1996.

[22] F. Zwicky, *Discovery, Invention, Research - Through the Morphological Approach.* Toronto: The Macmillan Company, 1969.

[23] F. Zwicky, “The morphological method of analysis and construction,” *Courant Anniversary Volume*, pp. 461 – 470, 1948, New-York.

[24] F. Zwicky and A. Wilson, *New Methods of Thought and Procedure: Contributions to the Symposium on Methodologies.* Springer, 1967.

[25] C. P. Frank, J.-G. Durand, A. Levy, F. Allair, and D. N. Mavris, “Design of an improved green taxiing system focused around the landing gear,” in *14th AIAA Aviation Technology, Integration, and Operations Conference*, 2014.

[26] P. Hollingsworth and D. N. Mavris, “A method for concept exploration of hypersonic vehicles in the presence of open & evolving requirements,” in *2000 World Aviation Conference*, 2000.

[27] W. O. Engler, P. T. Biltgen, and D. N. Mavris, “Concept selection using an interactive reconfigurable matrix of alternatives (irma),” in *45th AIAA Aerospace Sciences Meeting and Exhibit*, 2007.

[28] H. Jimenez and D. N. Mavris, “An evolution of morphological analysis applications in systems engineering,” in *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2010.

[29] D. N. Mavris, C. de Tenorio, and M. Armstrong, “Methodology for aircraft system architecture definition,” in *46th AIAA Aerospace Sciences Meeting and Exhibit*, 2008.

[30] O. J. Pinon, E. Garcia, and D. N. Mavris, “A methodological approach for airport technology evaluation and selection,” in *26th Congress of International Council of the Aeronautical Sciences (ICAS)*, 2008.

[31] L. R. Keller and J. L. Ho, “Decision problem structuring: Generating options,” *IEEE Transactions of Systems, Man, and Cybernetics*, vol. 18, no. 5, pp. 715 – 728, 1988.

[32] C. R. B. Arnold, R. B. Stone, and D. A. McAdams, “Memic: An interactive morphological matrix tool for automated concept generation,” in *Proceedings of the 2008 Industrial Engineering Research Conference*, 2008.

[33] O. J. Pinon, *A Methodology for the Evaluation and Selection of Adaptable Technology Portfolios and its Applications to Small and Medium Airports.* PhD thesis, School of Aerospace Engineering, Georgia Institute of Technology, 2012.

[34] Associate Administrator for Commercial Space Transportation (AST), “Reusable launch vehicle programs and concepts,” tech. rep., Federal Aviation Administration, 1998.

[35] F. Lehot, J.-F. Clervoy, F. Caquelard, R. Cauchois, P. Coue, G. Dodelin, F. Gai, R. Gucciardi, C. Lefevre, C. Mora, and P. Rosier, *Embarquer des Demain pour l’Espace.* Vuibert, 2010.

[36] J. C. Martin and G. W. Law, “Suborbital reusable launch vehicles and applicable markets,” tech. rep., The Aerospace Corporation: Space Launch Support Division - Space Launch Operations, 2002.

[37] X Prize Foundation, “Ansari X PRIZE.” Available online `http://space.xprize.org/ansari-x-prize` accessed 04.18.2014, 2011.

[38] BBC News, “The Astrium space jet.” Available online `http://news.bbc.co.uk/2/shared/spl/hi/pop_ups/07/sci_nat_the_astrium_space_jet/html/1.stm` accessed 04.21.2014, 2007.

[39] Scaled Composites, LLC, “Scaled Composites.” Available online `http://www.scaled.com/hires_gallery/gallery/X-Prize_1/single/XPrize_X1_0178` accessed 04.21.2014, 2014.

[40] Encyclopedia Astronautica, “Encyclopedia Astronautica: Black Armadillo.” Available online `http://www.astronautix.com/craft/bladillo.htm` accessed 04.21.2014, 2014.

[41] R. Chelouah and P. Siarry, “A continuous genetic algorithm designed for the global optimization of multimodal functions,” *Journal of Heuristics*, vol. 6, pp. 191 – 213, 2000.