David Manuel Judt , Craig Lawson , "Development of an automated aircraft subsystem architecture generation and analysis tool",
Engineering Computations, Vol. 33 Iss: 5, pp.1327 - 1352
Doi: 10.1108/EC-02-2014-0033

Development of an Automated Aircraft Subsystem Architecture Generation and Analysis Tool

David Manuel Judt and Craig Lawson,
Department of Aerospace Engineering, Cranfield University, Cranfield, UK

Structured Abstract:
Purpose - The work presents a new computational framework to address future preliminary design needs for aircraft subsystems. The ability to investigate multiple candidate technologies forming subsystem architectures is enabled with the provision of automated architecture generation, analysis and optimization. Main focus lies with a demonstration of the frameworks workings, as well as the optimizer's performance with a typical form of application problem.

Design/methodology/approach - The core aspects involve a functional decomposition, coupled with a synergistic mission performance analysis on the aircraft, architecture, and component levels. This may be followed by a complete enumeration of architectures, combined with a user defined technology filtering and concept ranking procedure. In addition, a hybrid heuristic optimizer, based on ant systems optimization and a genetic algorithm, is employed to produce optimal architectures in both component composition and design parameters. The optimizer is tested on a generic architecture design problem combined with modified Griewank and parabolic functions for the continuous space.

Findings - Insights from the generalized application problem show consistent rediscovery of the optimal architectures with the optimizer, as compared to a full problem enumeration. In addition multi objective optimization reveals a Pareto front with differences in component composition as well as continuous parameters.

Research Limitations - This paper demonstrates the frameworks application on a generalized test problem only. Further publication will consider real engineering design problems.

Originality/Value - The paper addresses the need for future conceptual design methods of complex systems to consider a mixed concept space of both discrete and continuous nature via automated methods.

Keywords: Subsystems, Aircraft, Design, Heuristic, Optimization, Automated

Article Classification: Research Article

This article is © Emerald Group Publishing and permission has been granted for this version to appear here https://dspace.lib.cranfield.ac.uk/handle/1826/10838?mode=full&submit_simple=Show+full+item+record. Emerald does not grant permission for this article to be further copied/distributed or hosted elsewhere without the express permission from Emerald Group Publishing Limited.

I. Introduction

In the context of aircraft conceptual design, relying on a database of design parameters from past vehicles is common practice when analysing new aircraft concepts [1][2][3]. By establishing trend lines for important variables, such as thrust to weight ratio or wing loading, a design to meet new mission requirements can be sized quickly and robustly. Utilizing these past concepts and relying on proven technologies reduces the risk associated with the project and improves the operational safety. Each newly evolved aircraft benefits from its predecessors design knowledge and aims to bring improvements, such as reduced fuel burn and structural mass.

However this conventionalism also tends to avoid advancement of revolutionary concepts and has resulted in familiar configurations of civil transport aircraft over the past five decades. Similarly in subsystem design such historical and empirical relationships also significantly limit the design space to existing and well documented technologies. If one wishes to venture away from the conventional, usually a number of "fudge factors" are employed to adjust the empirical trend lines.

The challenge engineers face during the initial phases of a design cycle is somewhat paradoxical; little is known about the problem, but full design freedom is available. The main focus is on exploring numerous potential solutions and concepts, while keeping them as changeable as possible. As concepts mature and analysis knowledge is gained, the commitment to these designs grows. Hence an increasing resource allocation takes place, making its costly to implement major design changes. Figure 1 demonstrates these trends for current and future design processes.

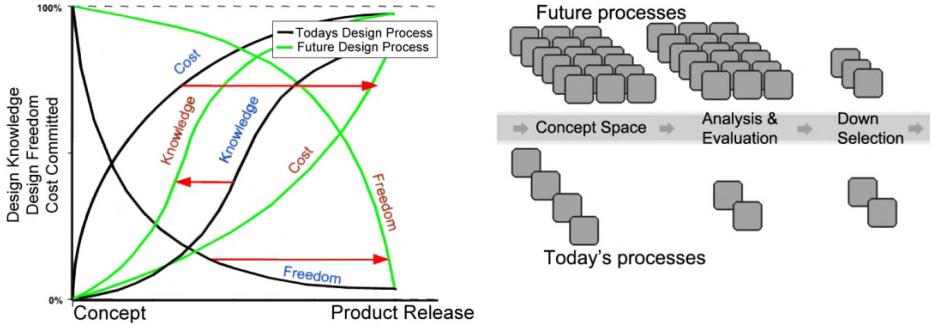


Figure 1: Current and future design processes based on [4]

Increased concept knowledge and keeping design options open, may delay the rise of committed cost during the design process. Such features require flexible design methods to deal with a multitude of potential solutions. Another challenge is to provide a framework which has low development time requirements when applied to new design problems.

In aircraft subsystem conceptual design there is generally more information available since the subsystem design cycle initiates later within the overall aircraft design process. First estimates of subsystem design metrics, such as mass and power requirements are often based on empirical equations established from design experience and past design data [5][6]. The underlying assumption implicit in the use of such a model is that the subsystem's architecture will be similar to past designs. This substantial restriction on the design freedom may not allow for the exploration of architectures which include novel technology components and different structural configurations.

Hence in the spirit of pushing trends as shown in Figure 1, a change in design approach is beneficial. An approach comparable to aircraft design's multi-disciplinary optimization (MDO) seems appropriate with adaptations specific to subsystem design. Such adaptations will accommodate tasks of generalizing subsystems to represent a wide concept space, with not only varying parameters, but also different architectures.
One technique for facilitating this generalized formation of architectures is the functional decomposition of the design problem [7][8][9]. These functional requirements are general enough to be linked to various physical component models, making up the architecture concept. Such a process, combined with an intelligent framework, of component filtering, functional reconfiguration and input/output variable linkages, is flexible enough to allow the consideration of a diverse range of architecture types [11]. Combined with model based approaches and moving away from empirical methods, an extended design space is available.
Past research has built parametric disciplinary models for the array of aircraft subsystems [12][13]. Using expert knowledge and physical relationships, power consumption, transformation, distribution as well as generation systems was modelled for both sizing and performance execution. Even though it proved beneficial for model based engineering simulations, the problem setup and concept generation was not automated.

A few other approaches looked specifically at automating the setup of the systems architecture models from individual component elements [14][15][16]. They incorporated a feedback between subsystem sizing and the aircraft level performance. This makes it possible to study and optimize subsystems for both aircraft and subsystem level objectives.
The other challenging element is the optimization strategy for the subsystem design space. Combining such new architecture flexible methods with gradient based optimizer methods can be difficult since the cost function is often nonlinear and non-uniform for changes in architecture analysis model structures. Past research have used heuristic based optimizers, such as genetic algorithms and ant colony optimization to deal with this mixed continues and discrete system design space.[18][19][20]

The presented work focuses on combining elements of architecture definition, multidisciplinary modelling, and design space optimization. Aspects of work by multiple authors are composed into a single framework and supplemented by novel method developments. This setup is able to achieve an improved cycle of concept design space expansion and contraction, as demonstrated via a theoretical design problem. Section II will describe the framework's theoretical processes and their integrated function. This includes a substantial subsection on the hybrid optimization procedure to make the reader familiar with its requirements in a mixed design space. Section III will demonstrate the frameworks optimizer performance with a theoretical architecture design problem comparable to an aircraft subsystem.

II. Framework Formulations

An overview of the framework is presented in Figure 2. Initially information is required to accurately define the system design problem and its setting in the overall aircraft concept. This establishes a clear boundary between elements considered part of the subsystem and other aircraft elements. Requirements, both functional and performance are extracted and associated to component models and operational scenarios.

With any approach to a design problem, the first step involves capturing the design requirements, from various sources. These could be received from the customer itself, often formulated in the form of performance requests, setting the design within the current and future marketplace.
With aircraft subsystem design requirements are also handed down from higher disciplines within the design hierarchy, such as structural and aerodynamic considerations. Additionally, certification and regulation standards play an important part in framing requirements appropriately.

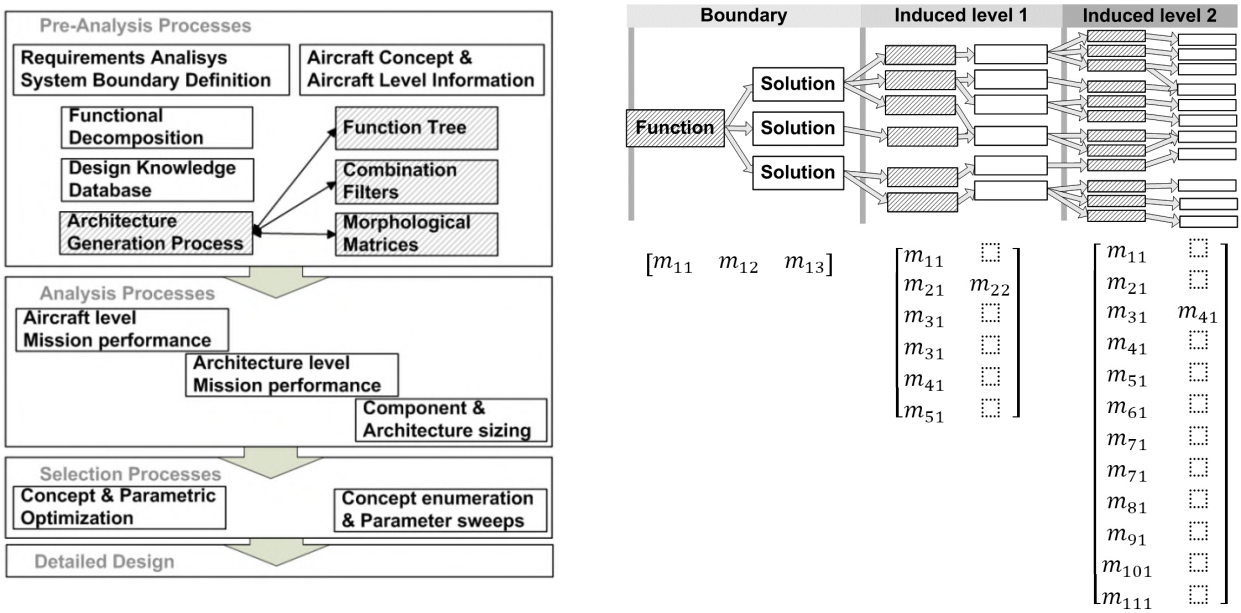


Figure 2: Framework outline & Function tree structure

A. Functional Decomposition and Knowledge Database

The captured requirements are linked to the main functionalities of the concept over its defined boundary. The boundary serves to differentiate the subsystem from the rest of the aircraft concept and the external environment. It also clarifies which interactions need to be considered between these regions during later analysis of design concepts. Boundary functions are linked to multiple possible component solutions which in turn induce further functional and performance requirements for the system concept.

Such a functional breakdown of the design problem generates a “tree structure” of possible solutions and functions [9]. It represents the broad potential space of unique assemblies of components forming the aircraft system concepts. A three level function struture is shown in Figure 2 together with its morpholical matrix representation.
Information related to the functional breakdown is collected in a knowledge database. Both compositional and connectivity information is required as it is used during the analysis methods. Additionally sizing scenarios need to be provided for the component options where appropriate. These scenarios drive the analysis procedure with a root finding mechanism to automatically assign the appropriate component parameters to achieve the specified performance targets at given operational conditions.

B. Enumeration Procedure

A complete database may contain prohibitively large numbers of functions and solutions for an enumeration. Hence two possible options are available for system architects. Either a down selection of component solutions takes place *a priori*, or the complete database is optimized with user defined objectives. The optimization option will be described in section E. An enumeration produces all unique combinations of component solutions present in the function tree, while taking into account user constraints to exclude infeasible technology combinations.
The matrix containing the possible solution choices is referred to as a morphological matrix, often encountered during idea generation in engineering design processes [21][22]. For a simple combinatorial analysis of a morphological matrix M of component solutions, the computation gives the matrix of combinations N with n_{cmb} columns, for A rows in M and B entries per row A.

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \quad N = \begin{bmatrix} m_{11} & m_{11} & m_{11} & m_{11} & \dots \\ m_{21} & m_{21} & m_{21} & m_{22} & \dots \\ m_{31} & m_{32} & m_{33} & m_{31} & \dots \end{bmatrix} \quad n_{cmb} = \prod_{j=1}^{A} \left\{ \sum_{i=1}^{B_j} m_{j,i} \right\}$$

Equation 1: Enumeration of a morphological matrix

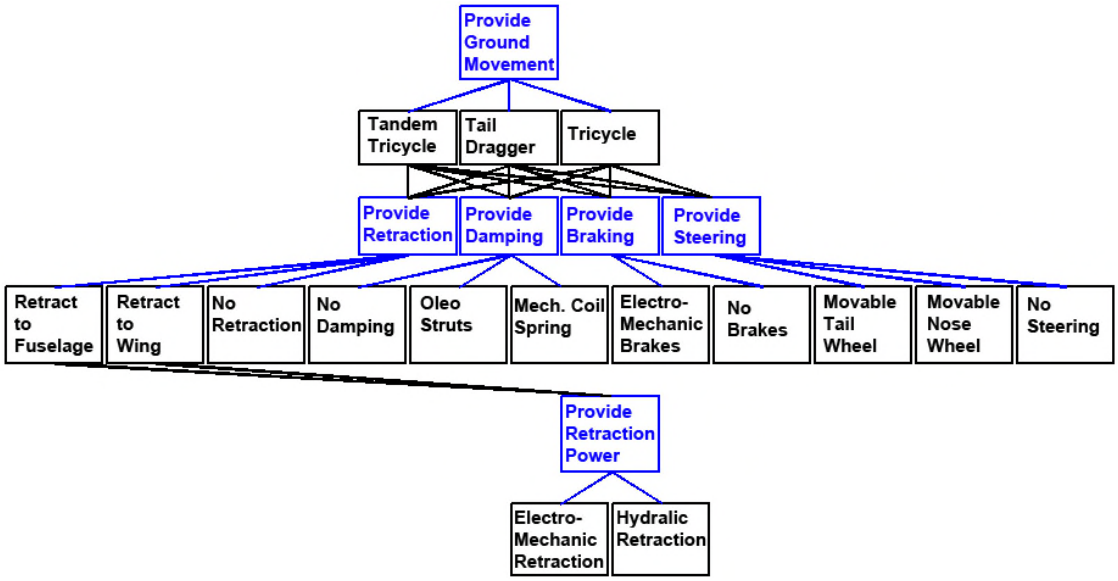


Figure 3: Function tree example for a landing gear system

$$M_1 = [Tandem, Tail\ dragger, Tricycle] \quad \rightarrow N_1 = [Tandem,\ Tail\ dragger, Tricycle]$$

$$M_2 = \begin{bmatrix} Retract\ to\ Fuselage & Retract\ to\ Wing & No\ Retraction \\ No\ Damping & Oleo\ Struts & Coil\ springs \\ Elec.Mech.Brakes & No\ Brakes & - \\ Mov.Tail\ wheel & Mov.Nose\ wheel & No\ steering \end{bmatrix} \rightarrow N_2 = \begin{bmatrix} Retract\ to\ Fuselage & Retract\ to\ Wing & No\ Retraction & \dots \\ No\ Damping & No\ Damping & No\ Damping & \dots \\ Elec.Mech.Brakes & Elec.Mech.Brakes & Elec.Mech.Brakes & \dots \\ Mov.Tail\ wheel & Mov.Tail\ wheel & Mov.Tail\ wheel & \dots \end{bmatrix}$$

$$M_3 = [Elec.Mech.Retraction \quad Hydraulic\ Retraction] \quad \rightarrow N_3 = [Elec.Mech.Retraction \quad Hydraulic\ Retraction]$$

$$N_{1,2,3} = \begin{bmatrix} Tandem \\ Retract\ to\ Fuselage \\ No\ Damping \\ Elec.Mech.Brakes \\ Mov.Tail\ wheel \\ Elec.Mech.Retraction \end{bmatrix}$$

Figure 4: Multi level enumeration for landing gear problem

The number of combinations may be computed with Equation 1 and depends on the dimension and number of elements in matrix *M*. This concept is extended to a multi branched function tree by decomposing it into a dynamic matrix structure. The combinatory approach is now applied in conjunction with the progression through the levels within the function tree, which implies a variation in matrix *M*. By executing the enumeration for each new Matrix *M,* constructed by the solution components satisfying the induced functions from the column entries in the previous results matrix *N,* the complete tree structure is considered. Figure 3 shows a system design example with two induced functions. The resulting changing morphological matrices *M* based on previous choices in the solution matrix *N* are shown in Figure 4. The total architecture combinations are then assembled from the individual *N* matrix entries providing the global collection of architectures enumerated in a component compositional sense.
Such a dynamic approach has a few advantages over past methods [7][8][9], where the function tree is rewritten as a large single morphological matrix, combined with an incompatibility matrix to exclude choice combinations. Due to the generation of individual matrices per level, their size is significantly smaller than a single global matrix. User defined combination filters will apply immediately during the enumeration, preventing the generation of infeasible results. Both of these two aspects allow a large problem to be enumerated with relatively small memory and computational requirements.

C. Analysis Process

The generated architectures that are feasible in terms of functionality are analyzed for their performance through performance modeling and sizing of all their constituted components individually. Component models should be of low to medium fidelity to keep execution times acceptable. As long as the interface conditions for each component model are satisfied, its content can be decided upon relatively freely by the technology experts. The database population and problem decomposition frequently produces updates in interfaces and variable exchanges as the design processes advances; hence the connectivity between models is reviewed continuously. Once a sufficient level of confidence is reached, the rest of the analysis will be fully automatic, spanning across the three main levels; aircraft, architecture and component. This concept is shown on the right side of Figure 5 with an example for an architecture level block diagram of the landing gear example on the left.

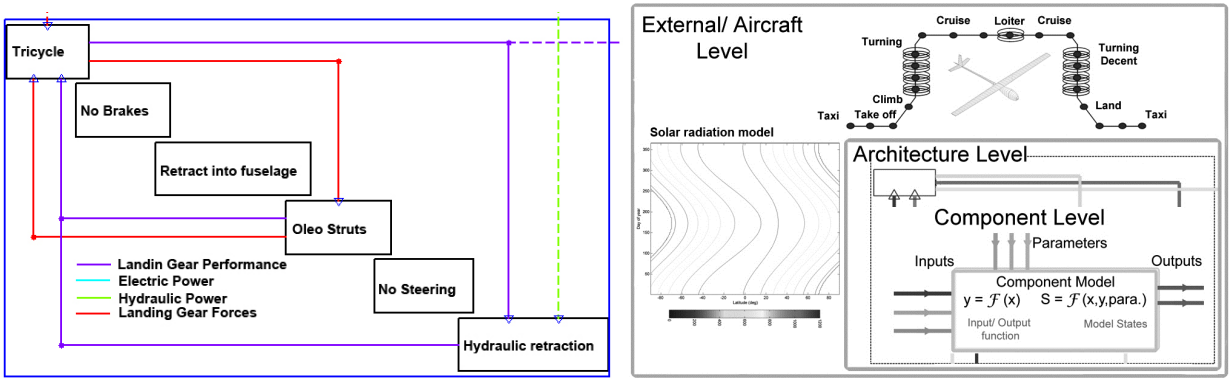


Figure 5: Analysis levels & landing gear example architecture diagram

Depending on the top level design requirements and the architectural interfaces with the external world, relevant mission profiles need to be provided. For a civil airliner, these may be built up from the conventional, taxi, take-off, climb, cruise, descent and landing segments. The flight profile is an input to the aircraft level models. Calculations on this level can be either empirical or analytical to determine the profiles of input variables to the relevant subsystem interfaces of the architectures. For the landing gear example the main aircraft level outputs will be wheel forces, moments and rotational velocities at current ground/flight conditions.
The analysis process on the architecture level starts by assigning initial values to the model and architecture parameters. The overall architecture model is then constructed from all individual component models with the use of the available connectivity information. Each component model is executed in turn, drawing on and modifying a global design structure matrix (DSM) of current input/output (I/O) conditions [10]. This is a fixed point iteration procedure, a common technique employed in multi-disciplinary analysis, tracking and computing models through the architectural setup. In the example architecture on the left side of Figure 5, example exchange variables include force interactions between the wheel and the damping component, the hydraulic power provision to the retraction mechanism and the performance of the gear.
Figure 6 shows an example of such a design structure matrix. Data in any given entry represents an input to its row ID and an output to its column ID. Hence each entry represents a connection between components. Depending on the type of connection, multiple variables may need to be passed to the next model; therefore entries are vectors of values.

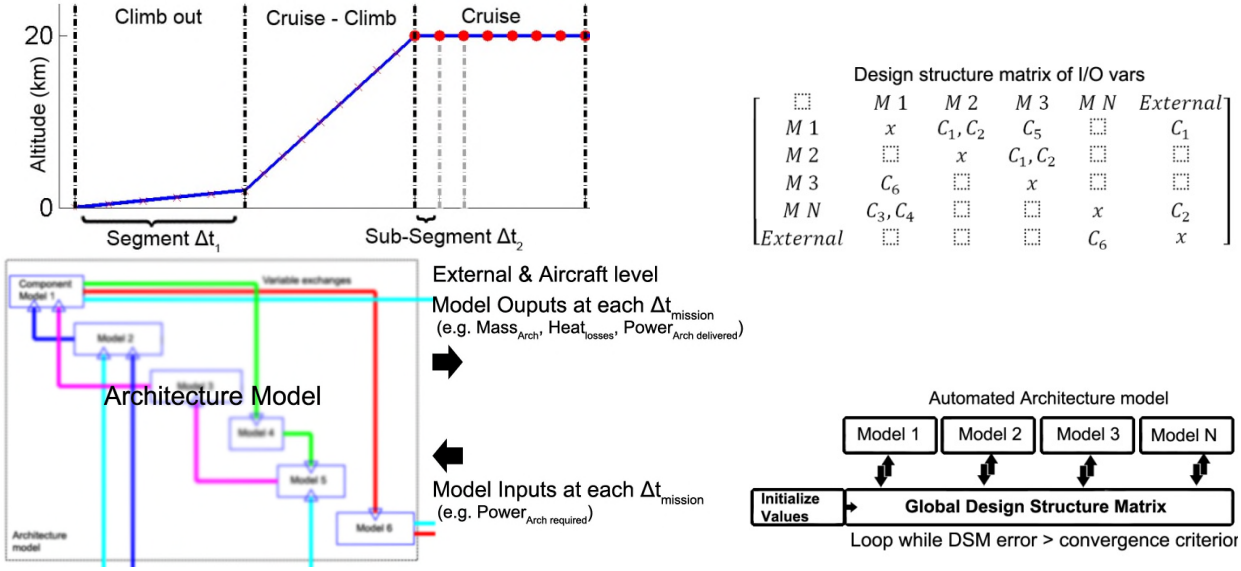


Figure 6: Fixed point iteration for DSM I/O balance

The architecture itself has I/O variables over its boundary to establish information flow to the aircraft and external levels. These values are assigned to the corresponding positions within the DSM, representing a single point performance condition of the architecture in steady state. The algorithm then computes the component models which vary their individual input conditions until the architecture flow of variables converges. Convergence is defined by the DSM error value which computes the summation of changes between all entries in the matrix after component model updates.

Once this iterative procedure converges it is repeated for each change in architecture I/O condition. This represents the performance of the architecture during a time varying mission profile, by the assuming that it can be modeled as a discrete set of steady state point performance results.

With the architecture mission performance evaluation, the reverse operation is undertaken, to size towards specific performance. In order to achieve this, re-computation of the architecture for sequentially adjusted parameters is necessary. However, individual component adjustment effects on performance are not separable from the architecture as a whole, making it necessary to follow a strategy which takes multiple targets into account.

The sizing process evaluates the component outputs and states for a representative mission and parameter settings. A root finding procedure then drives the parameter adaptation of a single component until a design performance is reached. Apart from the standard mission profile, the evaluation of mission critical failure scenarios also needs to be considered, since they size the backup components and systems. This process is repeated for all components, until there is no further change in sizing adaptations across all components for a single iteration as shown in Figure 7. The individual components, sized in equilibrium to each other, are a representation of the sized subsystem architecture. Due to the strong feedback between aircraft and architecture level models, further iteration is required to re-compute the changed mission profiles input to the architecture. Mass estimation and subsystem input power requirements tend to be the feedbacks of primary importance, however this may vary depending on the design problem at hand. This leads to a large update looping structure as shown in Figure 7, with the architecture variable balance per mission time step at the lowest level and the architecture concept at the highest.

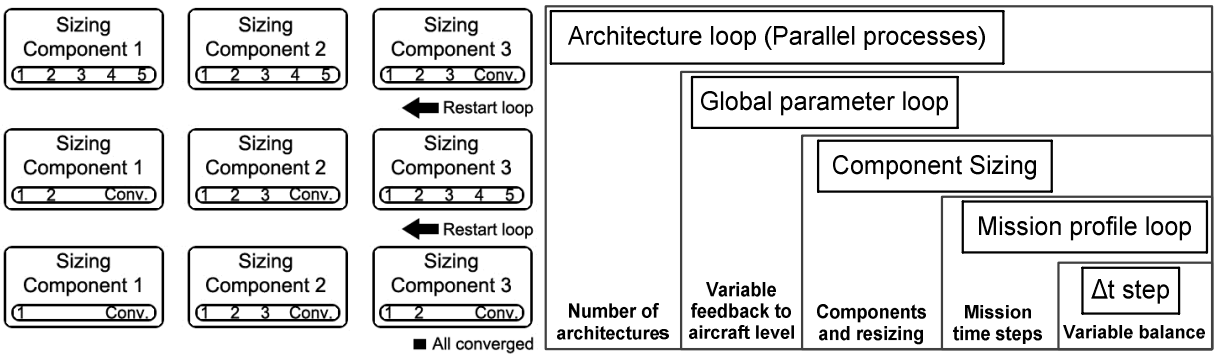


Figure 7 Sizing process with multiple sequential iterations & overall looping structure

Hence the fixed point iteration procedure described in Figure 6 takes place on the lowest two levels, followed by the component sizing and aircraft level feedback loop as shown in Figure 7. This represents the analysis & sizing of a single architecture concept, followed by the highest step, the architecture enumeration.

D. Ranking Process

The pool of enumerated architectures needs to be evaluated and down-selected to take a reduced set forward in the design process. Apart from choosing such solutions visually with the help of sliced trade-off diagrams, it is also possible to combine metrics to a single composite with the help of weighting factors.
After enumeration all feasible solutions are known and no re-evaluation computation is necessary. Hence a weighted objective method is quickly executed. A mathematical description of this weighted ranking procedure can be found in [11]. Once the concept space is reduced the remaining solutions can be re-computed for a range of component, architecture or aircraft level parameters. This provides insights for performance variations as well as other design metrics changes. Such parameter sweep scenarios help to evaluate concept competitiveness due to future development in technology values of subsystem components.

E. Optimization method

Alternatively to the enumeration and ranking an optimization method can be employed to identify best concepts. Additionally to the discrete concept space continuous parameters for components, mission profile and aircraft level also need to be included in the optimization. This yields a mixed continuous and discrete optimization problem which is effectively tackled with a hybrid combination of an ant systems optimizer and a genetic algorithm.
The function tree used to construct architectures can be represented as a graph problem, similar to the travelling salesman problem (TSP), illustrated in Figure 8. Solutions are constructed by passing through the required route points, with each choice affecting the competitiveness of the solution. Similarly the choice of components from the function tree, results in different objective function values of the constructed architecture.

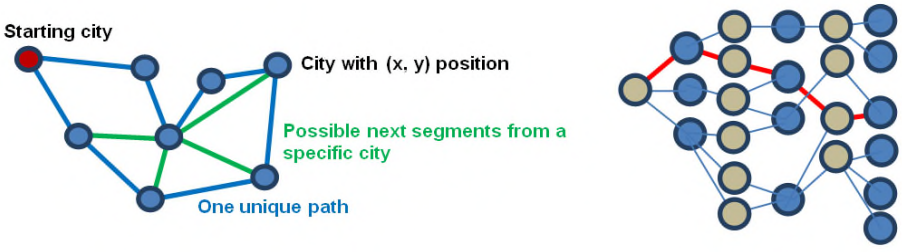


Figure 8: The TSP problem and a system architecture function tree equivalent

Such optimization problems have been successfully approached with the ant colony optimization (ACO) method [23][24]. It is particularly suitable to problems where discrete jumps in cost function are present, as well as a hierarchical problem choice structure. Individual ant agents initially move at random across the tree continuously depositing pheromones on path segments. Ants decide which route to follow based on the amount of pheromone they encounter. Hence a positive feedback effect emerges, leading to more and more ants following a given attractive path. At the same time all paths will slowly lose their pheromone levels through a simulated natural evaporation.
This behaviour is applied to a graph problem as shown in Figure 8 by utilizing multiple agents which generate paths through the function tree and deposit pheromone values on each segment of

the total path. At each junction in the function tree the segment choice is probabilistic with the likelihood given as a function of current pheromone values on each segment and the number of segment choices. Equation 2 shows this relationship for a segment probability p_{ij} from point *i* to *j*, with *N* total possible segments. The amount of pheromone deposition ΔPh_{Arch} is a function of the actual competitiveness of a path as evaluated by its objective function $Obj_{concept}$. This is the common ACO approach encountered in the literature [25]. The segment increment definition is shown in Equation 2 to Equation 6 which includes a separate operation for updates which have a lower objective value than all previously encountered paths. Such highly desirable paths then receive a boost equal to a multiple of the equilibrium pheromone level Ph_{equal}, assuring their higher probability in regeneration in subsequent ant choices. This is a framework specific modification to encourage faster convergence of strong concept paths. In order to prevent early overall convergence natural pheromone evaporation with rate Ev_{rate} is incorporated by reducing all pheromone matrix (PHM) values by a certain percentage after iterations, as shown in Equation 7.

$$p_{ij} = \frac{Ph_{ij}}{\sum_{j=1}^{N} Ph_{ij}}$$

Equation 2: Segment choice probability

If

$$\left(\frac{\min(Obj_{full\ set})}{Obj_{concept}}\right) > 1$$

Equation 3: Cut-off condition for pheromone boost

$$Ph_{equal} = \frac{\left(\frac{1}{1-Ev_{rate}}\right)}{\left(\frac{1}{1-Ev_{rate}}\right) - 1}$$

Equation 4: Boost pheromone increment

$$\Delta Ph_{Arch} = m * Ph_{equal}$$

else

$$\Delta Ph_{Arch} = \left(\frac{\min(Obj_{full\ set})}{Obj_{concept}}\right)$$

Equation 5: Standard pheromone increment

end

$$Ph_{path} = Ph_{path} + \Delta Ph_{Arch}$$

Equation 6: Addition of increment to path

$$Ph_{mat} = Ph_{mat} * (1 - Ev_{rate})$$

Equation 7: Evaporation of path

As an extension a population set of concept paths was considered during each generation, rather than single solution updates. The implementation is based on work presented in [26]. Each population set of generated paths is produced with the current pheromone values and then assessed for its best objective member path. This improves the effectiveness of the algorithm, since a more diverse range of solutions can be reached earlier in the process, eventually converging towards the optimal. Additionally, the separable nature of the population member computations, allows for parallelized executions. For multi objective applications of the ACO, a separate PHM is assigned to each objective, and updated according to the relevant objective

values. The probability of path segment choices is then produced with a weighted method of entries in all matrices as shown in Equation 8 for n_{Obj} Objectives with a weighting factor Wf_k.

$$p_{ij} = \frac{\sum_{k=1}^{n_{Obj}} Wf_k * \left(\frac{Ph_{ij}}{\sum_{j=1}^{N} Ph_{ij}}\right)_k}{\sum_{k=1}^{n_{Obj}} Wf_k}$$

Equation 8: Multi objective pheromone deposition with weighting factors

Coupled to this ACO method, which deals with the discrete architecture component choices, is a real value genetic algorithm (RGA). It utilizes aspects of survival of the fittest, gene crossover and a population pool as proposed in [27], to attain a global optimal parameter combination of the architecture values. The process begins by generating a set of random parameter combinations which forms the population's gene/trait pool. Once the population has achieved its desired size, breeding begins to generate offspring.
As a modification, the number of parents required for breeding can be user defined. The chosen parents are always the best members from the current population pool and breed with a further randomly generated candidate. Their parameter traits are then combined by probabilistic crossover to form one offspring. The amount of weighting for the random agent during the crossover determines the likelihood of inherited or mutated traits of parameters to be passed on to the offspring. A candidate in the GA population pool is a path choice in the function tree. Each of these paths contains certain parameter traits. For example the path shown in Figure 9 generates a solution of four components with three continuous component parameters:

$$\bar{p} = [\text{x12} \quad . \quad \text{x33} \quad \text{x44}]$$

Other paths will have different characteristics in the component composition resulting in other entries in the parameter vector.

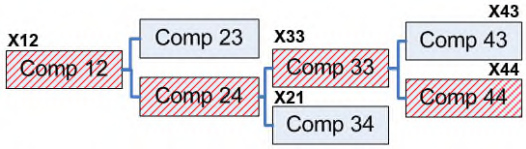


Figure 9: ACO solution path and its GA parameter traits

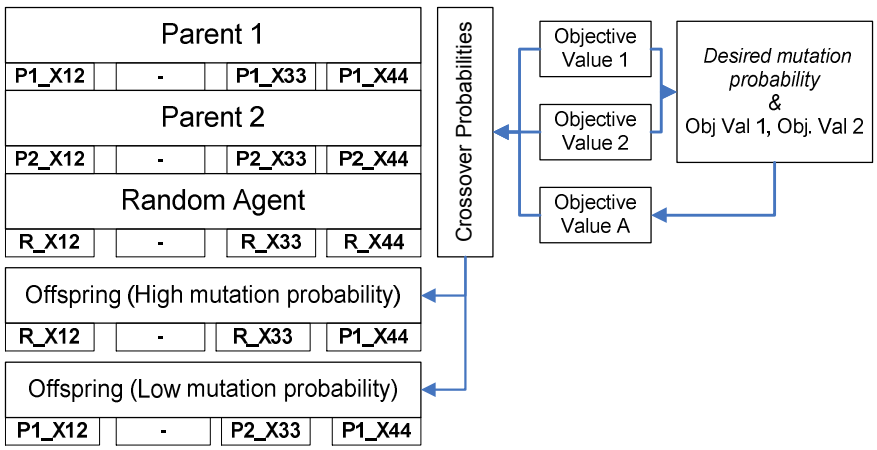


Figure 10: GA crossover and mutation process

For each parameter the probability of inheriting the value from one of the members of the breeding process depends on their objective value Ob_j as given by Equation 9. Figure 10 shows this process visually for 2 breeding parents and a random agent. The random agent uses the mutation rate β_{rate}, which is converted to an objective value by solving the equality in Equation 10 and the quadratic in Equation 11. Ob_A is the final objective value of the random agent to reach the desired mutation rate. Each breeding partner with the same parameter has the chance to pass on his parameter to the offspring, based on their objective values. Hence if high mutation rates are chosen random agents have a higher probability to pass on their parameters, due to the resulting high objective value as shown in Figure 9.

$$p_i = \frac{Ob_j}{\sum_{j=1}^{B} Ob_j}$$

Equation 9: Inheritance probability

$$\beta_{rate} = \frac{\frac{\sum_{j=1}^{B} Ob_j + Ob_A}{Ob_A}}{\left(\sum_{j=1}^{B} \frac{\sum_{i=1}^{B} Ob_j + Ob_A}{obj}\right) + \frac{\sum_{j=1}^{B} Ob_j + Ob_A}{Ob_A}}$$

Equation 10: Equality for random agent objective value

$$a = \sum_{i=1}^{B}\left(\beta_{rate} * \frac{1}{Ob_j}\right) \qquad b = \beta_{rate} + \sum_{j}^{B} \frac{\beta_{rate} * \left(\sum_{i=1}^{B} Ob_j\right)}{Ob_j} - 1 \qquad c = \left(\sum_{i=1}^{B} Ob_j * \beta_{rate}\right) - \sum_{i=1}^{B} Ob_j$$

$$ob_A = \frac{-b + \sqrt{(b^2 - 4 * a * c)}}{2 * c}$$

Equation 11: Quadratic to solve for random agent objective value

In addition to this offspring trait mutation through a random agent, a global mutation rate was introduced. This rate introduces further variability on all parameter traits of the offspring, through Equation 12.

$$P_{trait} = P_{trait} * \left(\left(1 - \beta_{glob}\right) + rnd_{(0-1)} * \left(2 * \beta_{glob}\right)\right)$$

Equation 12 GA global mutation

The breeding process is repeated to fully populate the required parameter of each path contained in the ACO pool. Each parent set only has a specific number of offspring at a time. This provides the ability for population members with poorer objective values to also induce some of their parameter traits to future candidates. Once a large enough set of offspring is generated, they are then incorporated into the current population, by replacing the equivalent number of dominated members of the set. The link between the ACO and GA algorithm, occurs during the path generation, which accounts for the past evaluated paths through the pheromone values. The GA then provides information about the parameter combinations which were successful in the past for the given concept paths. The resulting objective values are fed into the pheromone update function. Figure 11 shows the overall flow of processes of the hybrid optimizer.

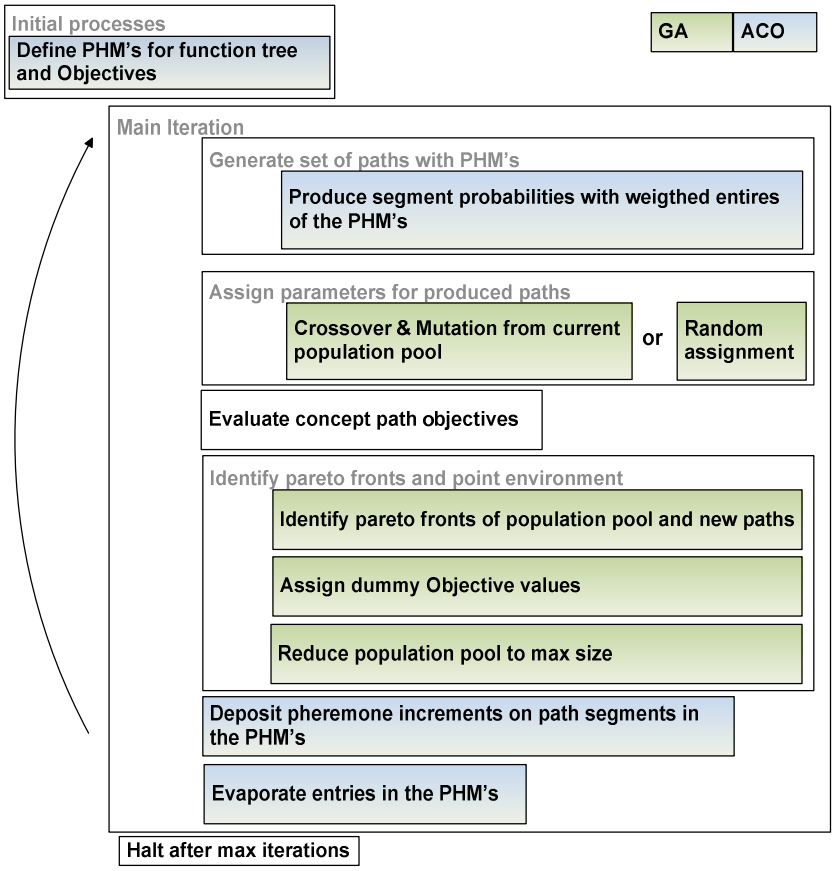


Figure 11: ACO-GA hybrid procedure

Since realistic engineering problems are trade-offs, a multi-objective optimization capability was deemed necessary. Specifically in aircraft systems design objectives such as weight, cost, power and efficiency are often conflicting and lead to a trade-off design. The adaptations to the hybrid approach include a non-dominant sorting method and assignment of a dummy objective based on the position and distribution of the discovered Pareto front points. This is a *posterior* approach resulting in Pareto fronts or surfaces, which can subsequently be used to select a solution with the most appropriate trade-offs between objectives, based on its shape.

The identification of the non-dominated points is done efficiently with a novel matrix method, presented here for a generalized n-dimensional space of data points as shown in Equation 13 to Equation 16. Consider an example of a two dimensional dataset P_{set} of n points.

$$P_{set} = \begin{matrix} x_1 & x_2 & x_3 & x_n \\ y_1 & y_2 & y_3 & y_n \end{matrix}$$

Equation 13 A two objective dataset

$$O_{ones} = \begin{bmatrix} 1 & 0 & 0 & \vec{n} \\ 1 & 0 & 0 & . \\ 0 & 1 & 0 & . \\ 0 & 1 & 0 & . \\ 0 & 0 & 1 & . \\ \overrightarrow{2n} & . & . & . \end{bmatrix} P_1 = \begin{bmatrix} x_1 & x_1 & \dots & \vec{n} \\ y_1 & y_1 & \dots & . \\ x_2 & x_2 & \dots & . \\ y_2 & y_2 & \dots & . \\ x_n & x_n & \dots & . \\ y_n & y_n & \dots & \vec{n} \end{bmatrix} P_2 = \begin{bmatrix} x_1 & x_2 & x_3 & x_n \\ y_1 & y_2 & y_3 & y_n \\ x_1 & x_2 & x_3 & x_n \\ y_1 & y_2 & y_3 & y_n \\ \dots & \dots & \dots & \dots \\ \overrightarrow{2n} & . & . & \overrightarrow{2n} \end{bmatrix}$$

$$\Delta D_{set} = P_1 - P_2$$

Equation 14 Difference equation

$$\Delta D_{idex} = \Delta D_{set} > 0$$

Equation 15 Differences to binary indices

$$Pa_{idex} = \sum_{j=1}^{nDim} \left(\Delta D_{idex}{}^{T} * O_{ones}\right)_j$$

Equation 16: Summation of indices to form non-dominated indices

The matrices P_1 and P_2 are reshapes of the data set to facilitate the matrix subtraction in Equation 14. Matrix O_{ones}, is populated by entries of ones along the diagonal, corresponding to the number of dimensions in the dataset. The subtraction represents the difference between all points in all dimensions. The results are then converted to a binary matrix ΔD_{idex}, which has entries of 1 for differences larger than zero and zero for others. By summing these indices over the dimensions, with a matrix multiplication of the transposed index matrix and O_{ones}, the indices of the points which are non-dominated Pa_{idex} are produced through a further summation over the number of dimensions of the datasets. Figure 12 shows this application, by evaluating a two objective data point set of randomly generated points in the interval 0.2 and 1.2.

In addition to the front identification, an environment measure of the point spread in the fitness assignment process is required. This is achieved by degrading fitness value Fit_i of each point i, by an amount equal to the reciprocal of the distances D_j between the test point P_i and its nearest neighbor P_j. Equation 17 shows this process for Np points in the test environment on the front.

$$\Delta Fit_i = \sum_{j=1}^{Np} \frac{1}{D_j}$$

Equation 17: Fitness degradation with root mean square distance between points

An even point distribution is achieved by degrading points, and hence moving the search direction towards more sparsely populated regions. The overall fitness assignment for the i_{th} point on Pareto front k is given by Equation 18.

$$Fit_i = \left(\sum_{j=1}^{Np} \frac{1}{D_j}\right) + M_k * k$$

Equation 18: Fitness assignment for combined front and spacing

The choice of optimization parameters is heavily dependent on the problem to be solved and required discretion by the user. Table 1 gives an overview of important parameters and their expected effects.

Table 1: Optimizer parameter effects

Parameter	*Global mutation rate*	*Cross over mutation rate*	*Population pool size*	*Parents for crossover*	*Evaporation rate*	*Paths per iteration*
Range	1 – 3%	15 – 30%	30 – 100	2 - 4	10 – 30%	4 – 8
Effects	Determines degree of randomness for all parameters	Determines degree of randomness per offspring	Determines max points on pareto front	Increases information sharing across pool	Increases random choices for paths in the ACO	Increases the search width per computation step

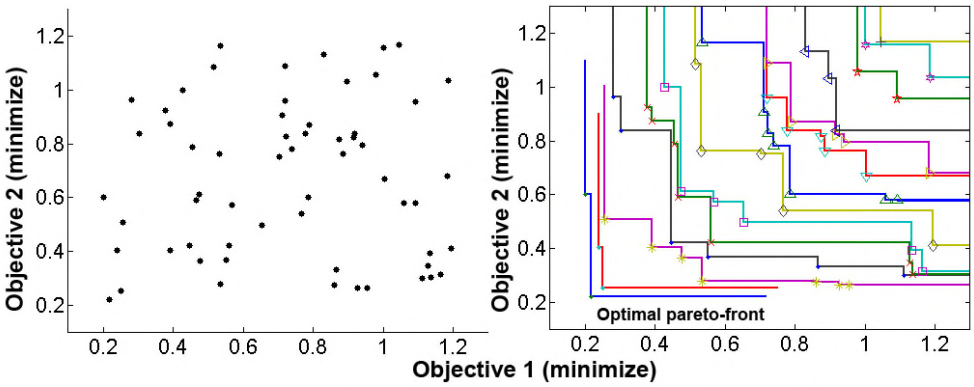


Figure 12: Pareto front identification of a random point cloud

III. Verification & Validation

The framework has been applied to aircraft subsystem design case studies [28][29], showing promise for large scale conceptual design space generation and selection. The intention of this section is to focus on a more generalized problem statement and testing the optimizer for the ACO, GA and the combined approach. The problem formulation is representative of a large scale design problem with multiple component alternatives and design metric implications.

One such relevant example may be the system power architecture of a civil passenger aircraft. Here various consumption devices are required from hydraulic actuators, electric compressors, fan heaters and fuel pumps to windscreen anti-icing. These are coupled on the system level to their power distribution components and the generation components. At the same time certain technology choices will lead to induced requirements to provide other component functionality. Hence the theoretical problem presented here, visualized by the function tree in Figure 13, is of somewhat similar nature.

The problem to be solved consists of a finding a composition of components N_{cmp} and component parameters, $P_{j,i}$ which minimizes metrics 1 and 2 (M_1 & M_2) for a whole architectures.

$$min_{(M_1,M_2)} \rightarrow [M_1, M_2]_{Arch} = f\left(N_{cmp}, NC_{cmp}, P_{j,i}\right)$$

Equation 19 Optimization test problem

$$[M_1]_{Arch} = \sum_{i=1}^{N_{cmp}} C_{1,i} * f_{1,cont}(P_i)$$

Equation 20 Metric 1

$$[M_2]_{Arch} = \sum_{i=1}^{N_{cmp}} \left[C_{2,i} * \left(\sum_{h=1}^{NC_{cmp}} C_{con_h} \right) * f_{2,cont}(P_i) \right]$$

Equation 21 Metric 2

Each component is assigned M_1 and M_2 and their summation forms the architecture optimization objectives to be minimized as given in Equation 19 . M_1 is a summation over all components in the architecture with each component value depending on a continuous function $f_{1,cont}$ and a constant $C_{1,i}$ as shown in Equation 20. For the component metric M_2 an increasing value with the component's connection number NC_{cmp}, connection constants C_{con} and continuous function $f_{2,cont}$ is chosen as shown in Equation 21. This makes M_1 comparable to a component mass metric due to component parameters and consequently overall system mass. M_2 changes both with component parameter and connections. Hence it incorporates two layers of information on the discrete level and is intended to be aligned to a complexity representation of an architecture.

The design problem representative function tree comprising 3 boundary functions with 2 induced function levels and a total of 32 components is used to establish the discrete design space. The tree shown in Figure 13 produces 9600 unique architectures after enumeration with the Pareto front visible and the minimal M_1 and M_2 architecture values at 201 and 318 respectively for fixed parameters. The concept space is clustered since architecture will have large number of common coponents, resulting in similar values for the metrics.

The continous functions apply to a subset of components, namely S1,S2, S4 and S7. This limits the analysis complexity of the problem by producing the component descrete space and a four dimensional continous space for two objectives.

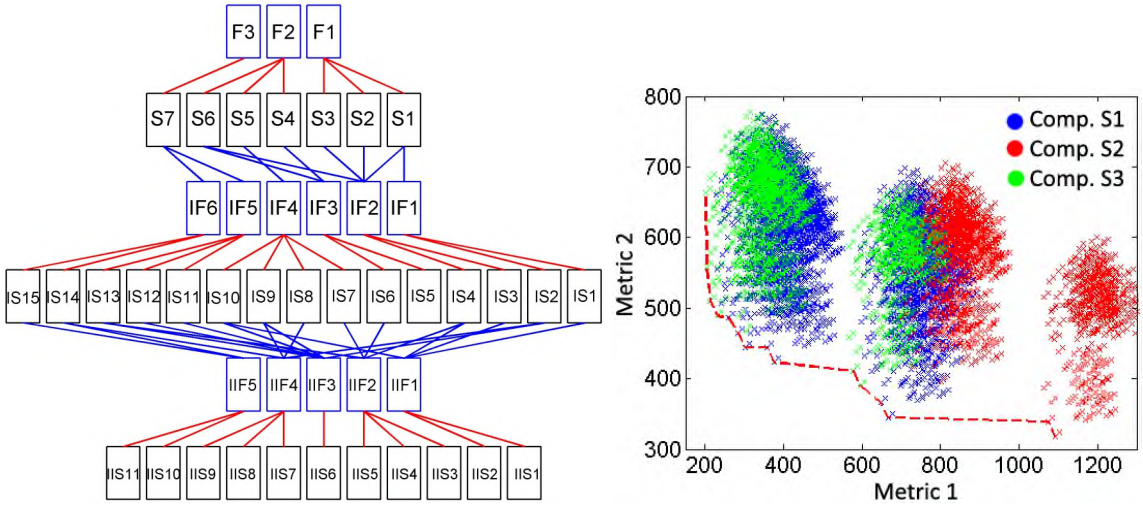


Figure 13: Function tree for the validation problem & categorized discrete space of solutions

Next the problem is optimized for a minimization of M_1 and M_2 individually, with fixed parameters for the continuous functions. Each iteration update considers a pool of 8 new candidates generated from the current pheromone map. Results of the converged architectures are shown in Figure 14, indicating the repeated convergence of the algorithm to the same solutions as discovered in the enumeration, with final M_1 and M_2 values at 201 and 318 respectively. Generally speaking as soon as the concept path with the lowest metric is discovered, the path element reinforcement leads to quick subsequent convergence. Results are obtained with the shown parameters in Table 2.

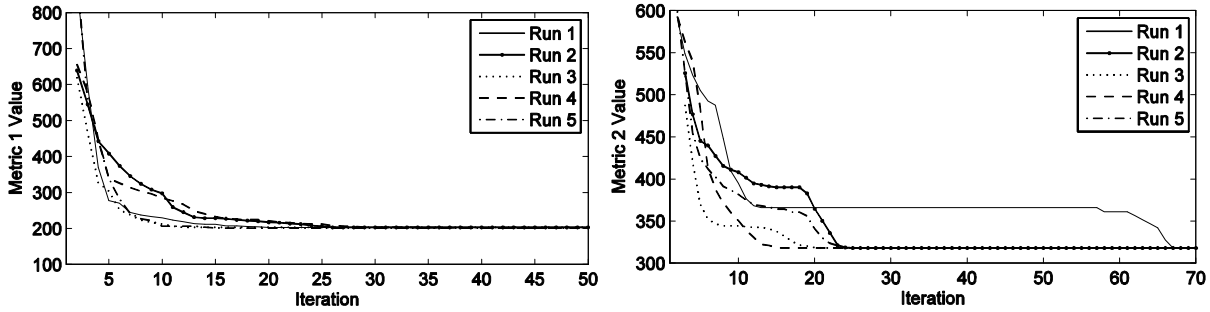


Figure 14: Metric averages in current population pool

Table 2: Single objective discrete space optimization parameters

Evaporation rate	Max Iterations	Paths per Iteration	Population pool size
15%	150	8	30

Next two objectives are considered and the Pareto front is compared with the result from the enumeration. Figure 15 shows five fronts from individual runs with a good agreement by the optimizer. Compared to the enumeration front shown in Figure 13, these results discover the front almost fully every time, with only minor differences. The results were produced with the optimization parameters given in Table 3.

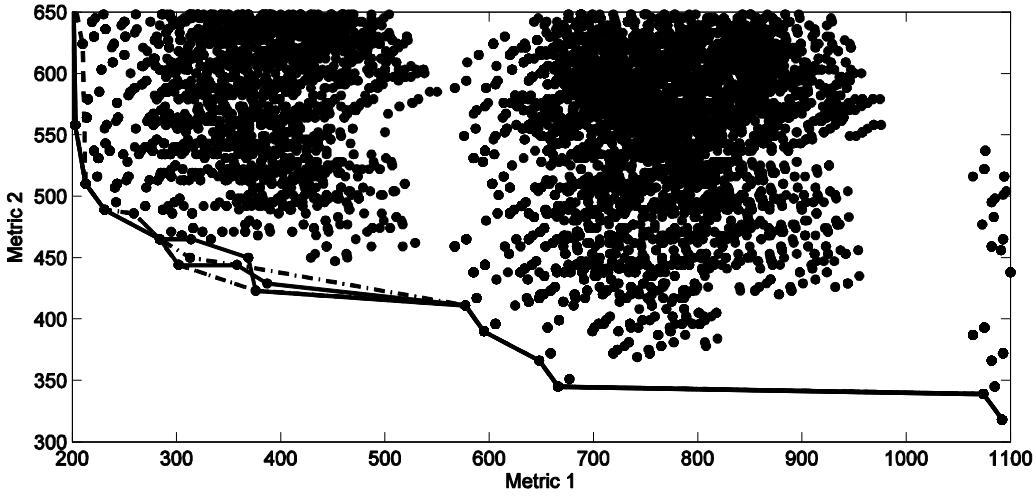


Figure 15: Pareto fronts and enumerated concept space

Table 3: Dual objective discrete space optimization parameters

Evaporation rate	Max iterations	Paths per Iteration	Population pool size	Objective M_1	Objective M_2
25%	400	8	30	Minimize	Minimize

The hybrid optimizer is next tested in combination with the discrete space. It is defined by parabolic and Griewank functions [24] of four parameters as shown in Equation 23 and Equation 24 respectively. The Griewank function induces multiple minima points in the M_2 direction; the parabolic has only one minimum in the M_1 direction. The global minima for f_2 is found when all parameters equal 2.5. The global minima for the parabolic function f_1, is encountered when all parameters equal 5. Function parameters are given in Table 4 and $nf = \frac{1}{\max(P_i)}$.

Table 4: Griewank and Parabolic function parameters

FR	Sc	O_x	O_y
90	30	-1	-2

$$f_1(P_i) = Sc * (nf * P_i + O_x)^2 + O_y$$

Equation 22: Single parametric parabolic function

$$f_1(\bar{P}) = \sum_{i=1}^{N_p} f_1(P_i)$$

Equation 23: Multi parametric parabolic function

$$f_2(\bar{P}) = \frac{\sum_{i=1}^{N_p}(P_i{}^2)}{FR} - \prod_{i=1}^{N_p}\left(\cos\left(\frac{P_i}{\sqrt{i}}\right)\right) + 1$$

Equation 24: Multi parametric Griewank function

An example extract of the functions is shown in terms of slices of the resulting 4D parameter space in Figure 16. Both functions have minima at different parameter values including multiple minima for the Griewank function. Hence the optimizer will search for global minima and best trade-offs between the parameters in the multi-objective optimization case.

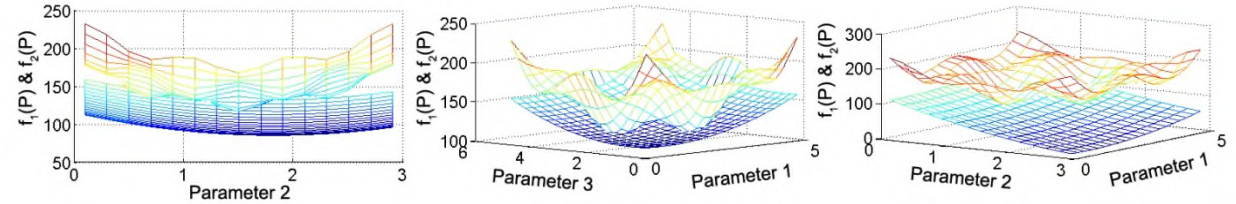


Figure 16: Continuous space slices for parameters

The results for a single objective optimization run of M_1 and M_2, for a fixed concept space are shown in Figure 17 & Figure 18. It repeatedly converges to the function minimum, with some variability in overall execution time. This variability is part of the probabilistic nature of the GA's mutation and crossover procedures. The optimization parameters are shown in Table 5.

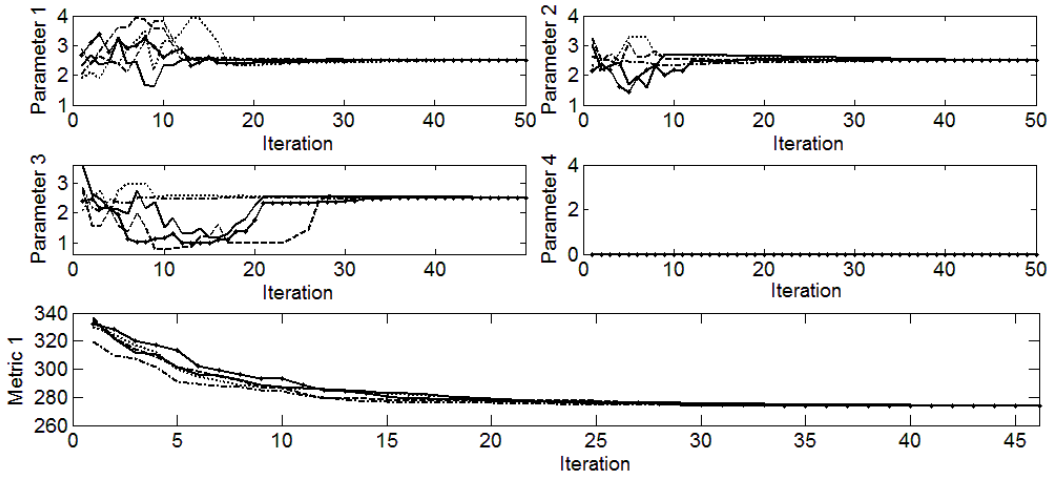


Figure 17: Single objective optimization for M_1; Convergence of parameters

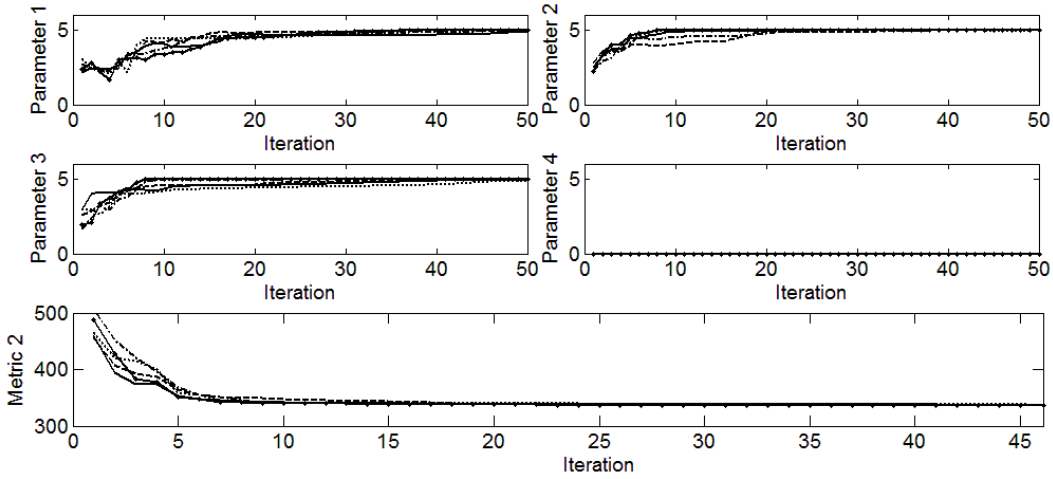


Figure 18: Single objective optimization for M_2; Convergence of parameters

Table 5: Single objective GA optimization parameters

Max iterations	Paths per Iteration	Population pool size	Global mutation rate	Cross over mutation rate	Parents for crossover
100	8	30	1%	15%	2

Finally the combined optimization problem in both parametric and concept space is solved. This means that metrics of concepts generated by the ACO will depend on the current component parameters which are varied by the continuous functions. The minimal solution to the single metric optimization will be the architecture with the following characteristics:

- Contains the fewest components possible with the lowest fixed $M_{1,2}$ component values
- Contains as many components with continuous function as possible, as these can reduce their fixed $M_{1,2}$ values further.
- Connection count component make within the architecture is at a minimum.

The following points are in some case conflicting, as some architecture will have more components, which have lower fixed metric values, or lower component connections. Results were produced with five repeated optimization runs for both metrics.

Figure 19 shows results for the ACO and GA combined M_1 optimization case. Repeated convergence to concept number 5187 is observable, which has three components of continuous function. The converged parameters for P_2, P_3 and P_4 produce the minimum for the Griewank function, as expected. It is visible that discovery of this concept can be slow, which is due to the fact that it differs only slightly in M_1 to a second concept with only two continuous functions.

Results for the M_2 case are shown in Figure 20. Here the final concept is number 5354 and contains three components with continuous functions. Again, parameters converge to values producing minima for the parabolic functions. Convergence is generally faster compared to the M_1 case. The final values for both M_1 and M_2 differ to the pure discrete optimization since the continuous functions reduce them further. Their minima are now at 179 and 211 respectively. The optimized parameters for this run are given in Table 6.

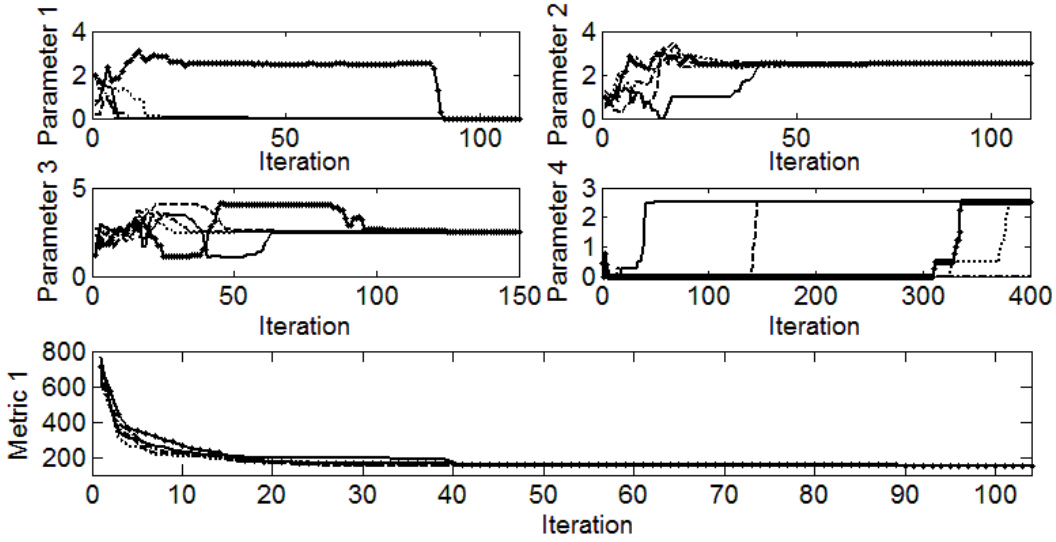


Figure 19: Parameter and objective data for the ACO-GA M_1 optimization

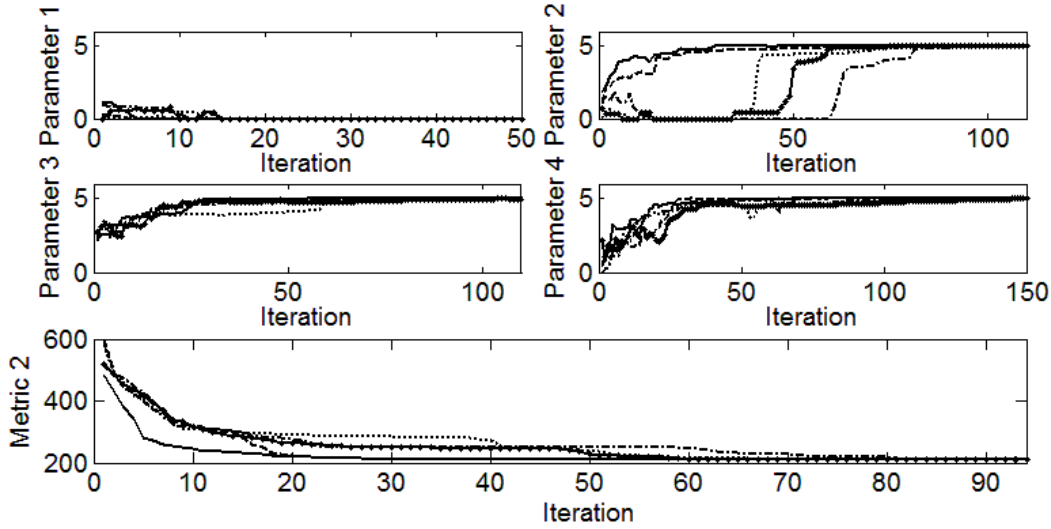


Figure 20: Parameter and objective data for the ACO-GA M_2 optimization

Table 6: Single objective GA and ACO optimization parameters

Iterations	Paths per iteration	Population pool size	Global mutation rate	Cross over mutation rate	Evaporation rate	Parents for crossover
400	8	100	1%	15%	35%	2

The resulting Pareto fronts for the two objective minimization optimization are shown in Figure 21. The coloring of the points along the front, correspond to the architectures in terms of their component composition. Their respective concept numbers have also been indicated. At the lower limit of the metrics the same architectures are discovered as was the case for the single objective case. Along the fractured front different concepts emerge with some variation in the discrete space. For repeated runs the discovered fronts show the same characteristics with minor differences in the parametric space. Figure 21 also shows the complete space of evaluated concepts during the run, colored according to the number of repetitions of the same concept number but with different parametric variables. Hence it shows the algorithm's ability to explore with direction towards the Pareto front. Discovered concepts are then evaluated and refined increasingly in those regions.

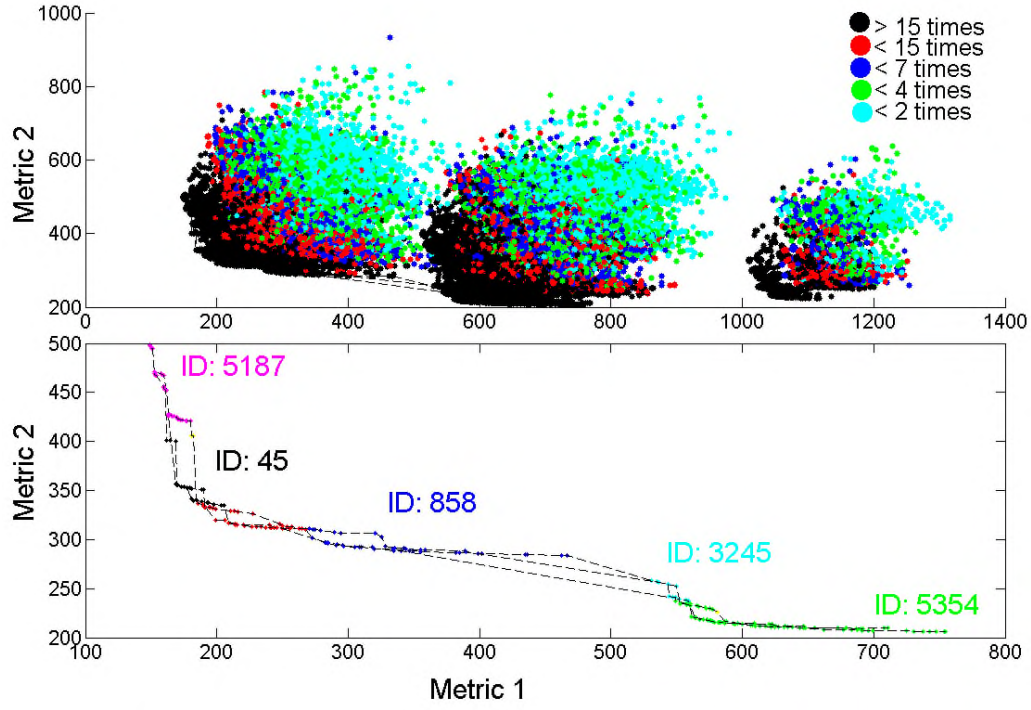


Figure 21: Pareto fronts for the ACO-GA dual objective optimization

IV. Discussion

The presented conceptual design framework represents a further step in automating the design processes, by including concept generation, analysis and optimization within a single environment. This brings the advantage of not having to handle data exchange between software packages and increases computational efficiency.

The theoretical concepts include a function based problem breakdown, technology exclusion filters, quasi-steady modeling of system component performance, clear sizing case definition for components and a mission model. The computing time is affected by the choice of model fidelity, mission simulation length and time increment size, as well as the optimization parameters. Hence for large architectures with many component models the computing requirements may become an issue as the analysis contains multiple loops and is subsequently coupled to the optimization iterations.

Common aspects of heuristic optimizers apply; a wider and more diverse search space will lead to longer convergence times. The focus on parallel computation of concepts provides a large benefit here, as the space can be wider with more computing resources and little impact on execution time.
On the one hand the theoretical concepts direct the framework towards aircraft applications with specific mission and aircraft models. Their interactions and interrelationships are hard coded in the computational routines. However, in theory the automated approach is applicable to a wider range of engineering problems, such as industrial plant and product design. One of the limitations of the framework is its assumption of establishing system performance based on component combinations, with little geometric considerations. Hence indirect effects such as installation requirements, power distribution efficiency factors and volume constraints are only considered in a qualitative manner. These indirect design objective variations will require consideration in future framework development, to ready it for large scale industrial applications such as the clean sky project [31].

V. Conclusions

A highly automated approach to conceptual engineering subsystem design has been presented. The population of a central database of design information and the effective combination of individual mathematical models of system components leads to automated generation and analysis of system architectures. Based on a functional decomposition of the design problem, component models are assembled and combined with their continuous design parameters to form a complex mixed design space. The developed ACO and GA hybrid optimizer is developed to handle this space.
A theoretical functional decomposition tree is combined with two continuous functions of known minima. Hence a comparison can be made for the enumerated architecture space with known minima of the parameter functions, against the hybrid optimizer. The structure of the presented theoretical study is comparable to aircraft conceptual design applications conducted by past research such as [18][19].
The optimizer discovers the expected minima for the known functions and the enumerated space, as well as the Pareto fronts. Further work is required to improve the confidence in the repeatability of results as well as the optimizer parameter settings such as mutation and pheromone evaporation rates.

References

[1] D. P. Raymer, *Aircraft Design: A Conceptual Approach*, AIAA Education Series, 4th Edition.

[2] D. Howe, "*Aircraft conceptual design synthesis*" Professional engineering publishing limited, London.

[3] D Mattingly, W. H. Heiser, D. T. Pratt., *"Aircraft engine design"*, AIAA Education Series, 2nd Edition.

[4] Blanchard, B., S., Fabrycky, W., J., *"Systems Engineering and Analysis"*, 3rd. Ed. Prentice-Hall, 1998

[5] I. Moir, A. Seabridge, *"Design and development of aircraft systems : An Introduction"*, Professional Engineering Publishing, 2nd Edition.

[6] I.Moir, A. Seabridge, *"Aircraft systems : mechanical, electrical, and avionics subsystems integration"*, Wiley, 3rd Edition.

[7] Mavris, D., Tenorio, C., Armstrong, M., "A Methodology for aircraft systems architecture definition", *46th AIAA Aerospace Sciences Meeting*, 7-10 January 2008

[8] Mavris, D., Tenorio, C., Armstrong, M., "Architecture Subsystem Sizing and Coordinated Optimization Methods", *47th AIAA Aerospace Sciences Meeting*, January 2009 in Orlando, Florida

[9] Armstrong, M., "A Process for function based architecture definition and modeling", *Msc. Thesis,* Georgia Institute of Technology, 2008

[10] Eppinger, S. D., Browing, T. R., "Design structure matrix methods and applications", MIT Press, 2012

[11] Judt D. M., Lawson, C. P., "Methodology for Automated Aircraft Systems Architecture Enumeration and Analysis", *AIAA ATIO Conference*, 2012

[12] Liscouet-Hanke, S., "A simulation framework for aircraft power systems architecting", *ICAS,*2008 Anchorage, Alaska

[13] Liscouet-Hanke, S., "A Model-Based Methodology for Integrated Preliminary Sizing and Analysis of Aircraft Power System Architectures" , *PhD Thesis*, Université de Toulouse, 2008

[14] Tenorio, C., "Methods for Collaborative Conceptual Design of aircraft power architectures", *PhD Thesis*, Georgia Institute of Technology, 2010

[15] J. Olvander, B. Lunden, H. Gavel, "A computerized optimization framework for the morphological matrix applied to aircraft conceptual design", *Computer-Aided Design 41*, 2009

[16] H. Gavel, J. Olvander, B. Johansson, P. Krus, "Aircraft fuel system synthesis aided by interactive morphology and optimization", *45th AIAA Aerospace Sciences Meeting*, 2007

[17] E. Safavi, V. Gopinath, J. Olvander, H. Gavel, "A Collaborative Tool for Conceptual Aircraft Systems Design", *AIAA Modeling and Simulation Technologies Conference*, 2012

[18] A. Chepko, O. Weck, D. Linne, E. Santiago-Maldonado, W. Crossley, "Architecture Modeling of In-Situ Oxygen Production and its Impacts on Lunar Campaigns", *AIAA SPACE 2008 Conference & Exposition*, 2008

[19] Pate, D., Patterson, M., German, B, "Methods for optimizing a family of reconfigurable aircraft", 11th*AIAA Aviation Technology, Integration and Operations*, 20-22 September 2011, Virginia Beach, VA.

[20] Villeneuve, F,"A Method for Concept and Technology Exploration of Aerospace Architectures", *PhD Thesis*, Georgia Institute of Technology, 2007

[21] Ritchey, T., “General Morphological Analysis” *16th EURO Conference on Operational Analysis, 1998*

[22] Zwicky, F., “Morphological Analysis and Construction”, *Wiley Inter-science*, New York, 1948.

[23] Angus, D., J., "Niching Ant Colony Optimisation" *PhD Thesis*, Swinburne University of Technology, 2008

[24] Buonanno, M., A., "A Method for Aircraft Concept Exploration Using Multicriteria Interactive Genetic Algorithms", *PhD Thesis,* Georgia Institute of Technology, 2005

[25] Dorigo, M., Maniezzo, V., and Colorni, A., "The ant system: Optimization by a colony of cooperation agent," IEEE Transactions on Systems, vol. 26 no. 1, pp. 1-13, 1996.

[26] Guntsch, M., Middendorf, M., "A population based approach for ACO", Proceedings of the Applications of Evolutionary Computing on EvoWorkshops, 2002

[27] Goldberg, D., E., "Genetic Algorithms in Search, Optimization & Machine Learning", Addison Wesley, 1989.

[28] Judt, D., M., "Methods for automated Aircraft Systems Architecture generation, Analysis and Selection", *PhD Thesis*, Cranfield University, Mai 2014

[29] Judt, D., M., Lawson, C., P., “Application of an automated aircraft architecture generation and analysis tool to unmanned aerial vehicle subsystem design”, *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering,* November 2014

[30] Griewank, A., O., "Generalized descent for global optimization", *Journal of Optimization Theory and Applications*, Volume 34, Issue 1, May 1981, Pages 11-39

[31] European Comission, “Clean Sky 2 Joint Undertaking Work Plan”, *www.http://ec.europa.eu, accessed 10/06/15*



https://dspace.org/

Development of an automated aircraft subsystem architecture generation and analysis tool

Judt, David M.; Lawson, Craig P.

2015-12-01
Emerald
http://dx.doi.org/10.1108/EC-02-2014-0033
David Manuel Judt, Craig Lawson, Development of an automated aircraft subsystem architecture generation and analysis tool, Engineering Computations, Vol. 33 Iss: 5, pp.1327 - 1352

Downloaded from a DSpace repository. https://dspace.org/