ADVANCES IN AIRCRAFT DESIGN: MULTIOBJECTIVE OPTIMIZATION AND A MARKUP LANGUAGE

by

Shubhangi Deshpande

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Computer Science and Application

December 5, 2013
Blacksburg, Virginia

Keywords: Multiobjective optimization, Pareto optimality, direct search method, surrogates, adaptive scalarization, Delaunay triangulation, multidisciplinary aircraft design, markup language, XML schema.

ADVANCES IN AIRCRAFT DESIGN: MULTIOBJECTIVE OPTIMIZATION AND A MARKUP LANGUAGE

by

Shubhangi Deshpande

(ABSTRACT)

Today's modern aerospace systems exhibit strong interdisciplinary coupling and require a multidisciplinary, collaborative approach. Analysis methods that were once considered feasible only for advanced and detailed design are now available and even practical at the conceptual design stage. This changing philosophy for conducting conceptual design poses additional challenges beyond those encountered in a low fidelity design of aircraft. This thesis takes some steps towards bridging the gaps in existing technologies and advancing the state-of-the-art in aircraft design.

The first part of the thesis proposes a new Pareto front approximation method for multiobjective optimization problems. The method employs a hybrid optimization approach using two derivative free direct search techniques, and is intended for solving blackbox simulation based multiobjective optimization problems with possibly nonsmooth functions where the analytical form of the objectives is not known and/or the evaluation of the objective function(s) is very expensive (very common in multidisciplinary design optimization). A new adaptive weighting scheme is proposed to convert a multiobjective optimization problem to a single objective optimization problem. Results show that the method achieves an arbitrarily close approximation to the Pareto front with a good collection of well-distributed nondominated points.

The second part deals with the interdisciplinary data communication issues involved in a collaborative mutidisciplinary aircraft design environment. Efficient transfer, sharing, and manipulation of design and analysis data in a collaborative environment demands a formal structured representation of data. XML, a W3C recommendation, is one such standard concomitant with a number of powerful capabilities that alleviate interoperability issues. A compact, generic, and comprehensive XML schema for an aircraft design markup language (ADML) is proposed here to provide a common language for data communication, and to improve efficiency and productivity within a multidisciplinary, collaborative environment. An important feature of the proposed schema is the very expressive and efficient low level schemata. As a proof of concept the schema is used to encode an entire Convair B58. As the complexity of models and number of disciplines increases, the reduction in effort to exchange data models and analysis results in ADML also increases.

ACKNOWLEDGEMENTS

First and foremost, I thank my advisor, Dr. Layne Watson, for being a guiding beacon through the long course of graduate studies that has not been an easy ride. His patience, encouragement, and immense knowledge were key motivations throughout my PhD. I would not have imagined having a better advisor and mentor for my research. Similarly, I would like to thank Dr. Robert Canfield for being such a great mentor and coadvisor, for letting me be a part of the SensorCraft group, and for providing insightful and timely feedback on this research. I would like to thank Drs. Ramakrishnan, Fox, Kolonay, and North for serving on my thesis committee and giving me advice. I would like to thank Dr. Blair for his guidance and support during the early stage of this dissertation. Furthermore, I thank all the CCMS aficionados for their continued help and support through this journey. I would like to thank Anthony and David for providing the use cases for the ADML work. Many thanks to Nathan Love for his time, effort, and help in shaping the current version of the ADML schema. I would like to thank Rob Hunter and Jonathan Spence for their administrative help. I would also like to thank members of the academic staff for advice and support.

The most enjoyable part of this PhD has been meeting a wonderful group of friends. I thank Ted, Jingwei, and David for being great labmates, and making it a convivial place to work. I would like to thank Isha, Kartiki, Kalyani, Kshitija, Shvetha, and Mandar for being a memorable part of my graduate life. They have been like surrogate family, bearing the brunt of the frustrations, and sharing in the joy of the successes.

I would like to acknowledge my family, where the most basic source of my life energy resides. I would like to thank my parents and my sisters for their unconditional love and support. Mom especially has always been a role model of resilience, strength, and patience. I would like to thank my parents-in-law for their consideration and support.

Finally, I would like to thank my long suffering better half, Santosh, who endlessly encouraged me, and put up with the unavoidable side effects of doing a PhD in parallel with attempting to have a life. Words cannot express my gratitude for everything you have done. Thank you for accompanying me on this adventure, I look forward to our next one!!

This material is based on research sponsored by Air Force Research Laboratory under agreement number FA8650-09-2-3938. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1.

INTRODUCTION

The former multidisciplinary optimization branch (MDOB) at NASA Langley research center (LaRC) [34] identified frameworks for multidisciplinary analysis and optimization research, and set ambitious targets for designing future air vehicles. This vision promotes advanced integrated and collaborative analysis and design capabilities as key enablers, and asserts a multidisciplinary approach as being instrumental for achieving improved designs.

Since aircraft design became a highly multidisciplinary, collaborative endeavor, the process of conceptual aircraft design has changed tremendously in the past few decades, aided by rapidly developing computer technology. Analysis methods that were once considered feasible only for advanced and detailed design are now available and even practical at the conceptual design stage. Conceptual design, being primarily a search process, requires an extensive exploration of the design space in order to gain insight into the relation between the design variables and aircraft performance. Engineers within several diverse disciplines including but not limited to geometry, structures, aerodynamics, controls, propulsion, and flight mechanics perform iterative parametric evaluations until an optimal design is developed.

This expensive and time consuming multidisciplinary conceptual design process requires advanced technology and sophisticated tools for automation of interdisciplinary communication thereby decreasing design time. Multidisciplinary design optimization (MDO), used popularly for the design of complex engineering systems and subsystems, coherently exploits mutual interactions between the entities involved. This succinctly describes the multidisciplinary conceptual aircraft design and analysis process. A typical MDO problem involves several conflicting objectives that need to be optimized simultaneously; this process is popularly known as multiobjective optimization. For a nontrivial multiobjective optimization problem with p conflicting objectives no single solution can optimize all p objectives simultaneously. Instead, a set of optimal trade-offs known as Pareto optimal solutions is obtained; the goal of a practical multiobjective optimization method is to efficiently generate a good collection of well-spread Pareto optimal solutions. The classic approaches for solving multiobjective optimization problems are based on some scalarization (aggregating all the objectives into a single objective function and solving a MOP as a single objective optimization problem) technique, such as the weighted sum method or its variants, ϵ-constraint method, etc. The most commonly used scalarization approach is the convex combination method where the multiobjective optimization problem is converted to a single objective optimization problem using a predefined weight vector. A major drawback of the convex combination method is that, if the Pareto front has a nonconvex (or nonconcave) region, then the method fails to produce any points in that region. In real design problems a decision maker might be able to consider only a subset of the available solution set. In this context, having a good representation of the entire Pareto front (producing solutions in all parts of the Pareto front with a good distribution of the solutions along the front) is a crucial requirement in order to make an informed decision. There are very few other techniques for deterministically solving MOPs (such as normal boundary intersection and its variants) that address the issue of efficient generation of well-represented Pareto optimal fronts.

Another technology gap in the current conceptual aircraft design and analysis process is the lack of a standard for interdisciplinary data communication. The changing philosophy for conducting conceptual aircraft design and analysis poses additional challenges beyond those encountered in a low fidelity design and analysis of aircraft. Typically, each entity involved in a multidisciplinary aircraft design process both consumes (input) and produces (output) a large amount of data. Efficient manipulation and exchange of this disciplinary data among different entities (platforms, users, application codes, etc.) demands a standard data format. Although the use of sophisticated design and analysis tools has become prevalent in the aerospace community, the field of interdisciplinary communication still remains in a primitive state.

This thesis takes some steps towards bridging the gaps in existing technologies and advancing the state-of-the-art in conceptual aircraft design. Two computational tools are proposed here for the overall advancement of multidisciplinary conceptual aircraft design and analysis. In particular, the first part of the thesis proposes a new multiobjective optimization method that combines ideas from simplical topology and computational geometry and achieves a reasonably close approximation to the Pareto front with well-distributed optimal solutions for diverse types of multiobjective optimization problems. The ultimate goal is to apply the proposed algorithm to complex MDO problems in conceptual aircraft design and analysis. The second part of the thesis deals with interdisciplinary data exchange issues, and proposes an XML schema based markup language ADML (aircraft design markup language) for data communication in a multidisciplinary, collaborative conceptual aircraft design and analysis environment.

The organization of this thesis is as follows. Chapter 2, the first part of the thesis, provides a review of existing literature, describes the proposed multiobjective optimization algorithm, and presents numerical evaluation of the proposed method on a set of test problems. Chapter 3, the second part of the thesis, presents the XML schema based markup language ADML. Chapter 4 offers concluding remarks. The complete ADML schema is provided in Appendix A.

Chapter 2.

MULTIOBJECTIVE OPTIMIZATION USING AN ADAPTIVE WEIGHTING SCHEME

2.1. Introduction

Multiobjective optimization problems (MOPs) arise in practically every area of science and engineering where optimal decisions need to be made in the presence of two or more conflicting objectives. A decision maker has to take into account different criteria that need to be satisfied simultaneously.

Let $\mathbb{R}^n$ denote real n-dimensional Euclidean space. For vectors $x^{(1)}$, $x^{(2)} \in \mathbb{R}^n$ write $x^{(1)} < x^{(2)}$ $(x^{(1)} \leq x^{(2)})$ if $\forall i$, $x_i^{(1)} < x_i^{(2)}$ $\left(x_i^{(1)} \leq x_i^{(2)}\right)$, $i = 1, \ldots, n$. Let l, $u \in \mathbb{R}^n$ with $l < u$ and $B = \{x \in \mathbb{R}^n \mid l \leq x \leq u\}$. Given p conflicting objective functions $f_i : B \to \mathbb{R}$, $i = 1, \ldots, p$, the multiobjective optimization problem is to simultaneously minimize all the f_i over B.

For a nontrivial multiobjective optimization problem with p conflicting objectives no single solution can optimize all p objectives simultaneously. Hence, a new notion of optimality, known as Pareto optimality (the set of optimal solutions is known as the Pareto front), is generally used in the context of multiobjective optimization problems. Pareto optimality is generally defined using a dominance relation as follows: Given two decision vectors $x^{(1)}$ and $x^{(2)}$, $x^{(1)}$ is said to dominate $x^{(2)}$, denoted as $x^{(1)} \prec x^{(2)}$, if and only if $x^{(1)}$ is no worse than $x^{(2)}$ in all objectives, and $x^{(1)}$ is strictly better than $x^{(2)}$ in at least one objective. A point $x^* \in B$ is a globally Pareto optimal solution if and only if there does not exist any x in the design space B such that $x \prec x^*$. A point $x^* \in B$ is a locally Pareto optimal solution if and only if for some $\epsilon > 0$ there does not exist any x in the neighborhood $\left\{x \mid \|x - x^*\| < \epsilon\right\} \cap B$ of x^* that dominates x^*.

The goal of a MOP is to find all the Pareto optimal solutions for all the conflicting objectives. In general, finding infinitely many solutions on a Pareto front is impossible, and finding even a large discrete subset is hard when the structure and/or the analytical form of the underlying objective functions is not known or is very complex (e.g., blackbox simulation optimization). An approximation to the true Pareto front is obtained as an alternative in such situations. The most common approach is to approximate the Pareto front by a discrete set of points. The other difficulty is the efficient generation of well distributed Pareto optimal solutions. In real design problems a decision maker might be able to consider only a subset of the available solution set. In this context, having a good representation of the entire Pareto front is a crucial requirement in order to make an informed decision. Efficiency (in terms of the total number of true function evaluations) of a multiobjective optimization algorithm is a key factor in practical multidisciplinary design optimization problems where a design cycle involves time consuming and expensive computations for each discipline. For example, in a conceptual aircraft design process several different disciplines influence each other resulting in several interdisciplinary iterations to reach a feasible optimal design. The task becomes even more complicated when the objective functions are evaluated using expensive blackbox simulations where analytical form of the objective functions is not available or is very complex.

The aim of this thesis is to propose a new multiobjective optimization method that provides a well distributed representation of the Pareto front for multiobjective blackbox optimization with a minimal number of true function evaluations. The general idea is to systematically move towards the Pareto front at the end of every iteration by adaptively filling the gaps between the nondominated points obtained so far. The biobjective optimization method proposed in [18] and [19] by the authors was based on the scalarization scheme of Ryu et al. [43]. However, the same adaptive weighting scheme does not work for multiobjective problems with $p > 2$ objectives. The algorithm presented in this thesis is an alternative approach for generalizing the adaptive weighting scheme of Ryu et al. [43] to problems with more than two objectives. The algorithm employs a hybrid approach using two derivative free direct search techniques — a deterministic global search algorithm DIRECT [30] for global exploration of the design space and a local direct search method MADS (mesh adaptive direct search) [5] to exploit the design space by fine tuning the potentially optimal regions returned by DIRECT and to speed up the convergence. Inexpensive surrogates for the objective functions are used in order to minimize the number of expensive simulation runs. The proposed method uses the global search algorithm DIRECT as a sampling method instead of using traditional random sampling (e.g., Latin hypercube sampling) or factorial designs that are expensive in higher dimensional design spaces. Results show that the proposed method provides well distributed Pareto optimal points on the Pareto front for diverse types of problems. Another contribution is the use of a precise mathematical measure, known as star discrepancy, for the distribution of the Pareto optimal solutions. A biobjective version presented in [18] has been generalized in the present work for problems with more than two objectives. A detailed description is provided in Section 4.1.

2.2. Background

Many different methods have been suggested in the literature for solving multiobjective optimization problems. The solution approaches can be divided into two broad categories — evolutionary approaches and scalarization or aggregation approaches. Applying a Pareto based ranking scheme using genetic algorithms has become very common and popular in most of the evolutionary approaches (e.g., [16]), although some schemes are based on particle swarm intelligence and simulated annealing. An evolutionary approach yields a set of nondominated points at the end of each iteration, requires a very large number of true function evaluations, and does not guarantee optimality in general. Aggregate or scalarization approaches [22] are considered to be classical methods for solving multiobjective optimization problems. The general idea is to combine all objective functions to convert a multiobjective optimization problem into a single objective optimization problem. Thus each iteration yields a single solution by solving a single objective optimization problem. The most commonly used scalarization approach is the convex combination method where the multiobjective optimization problem is converted to a single optimization problem using a predefined weight vector. A major drawback of the convex combination method is that if the Pareto front has a nonconvex (or nonconcave) region then the method fails to produce any points in that region. Also uniformly distributed weights may not produce uniformly distributed optimal points on the front. To alleviate this problem an adaptive weighting scheme was suggested by Ryu et al. in their biobjective optimization algorithm PAWS [43]. PAWS has an efficient weighting scheme, however, the central composite design based sampling strategy used in PAWS is impractical in higher dimensional search domains. Deshpande et al. [18] and [19] proposed an algorithm for biobjective optimization that employs the adaptive weighting scheme of PAWS, but overcomes the limitation of PAWS with a novel sampling strategy and a global search method. The ordering property exploited in these algorithms ([18] and [43]) for the scalarization of the objectives does not work in higher dimensional objective spaces (i.e., problems with more than two objectives). Hence, an alternative strategy is proposed in this thesis that works for problems with any number of objectives.

Other methods like normal boundary intersection (NBI) [14] and its variations solve multiobjective optimization problems by constructing several aggregate objective functions. A scalarization method called BiMADS [7] solves a biobjective optimization problem through a series of single objective formulations using the direct search method MADS, and attempts to obtain a uniform coverage of the Pareto front, even in the case when the Pareto front is nonconvex or disjoint. However, results produced using BiMADS are dependent on the random sampling used in the algorithm (e.g., Latin hypercube sampling used for the very first run in NOMAD 3.5.1 may not produce consistent results across different computing systems), and BiMADS also exploits the ordering property available in the two–dimensional case, and hence is not applicable for problems with more than two objectives. Audet et al. suggested an alternate formulation in their algorithm MultiMADS [7] to generalize the scalarization scheme of BiMADS to problems with more than two objectives. MultiMADS was proposed with an intention to overcome some of the issues in the NBI method, and could be considered as a state-of-the-art in the classical deterministic approaches for solving MOPs. MultiMADS (and BiMADS for the biobjective case) is used as a benchmark for the numerical evaluation of the proposed method.

Several real world scientific and engineering applications require analysis of spatially distributed data from expensive experiments or complex computer simulations that can take hours to run even on a fast workstation. This makes it difficult to explore and produce well distributed design combinations in high dimensional design spaces. Identifying the regions that contain good designs in such data-scarce domains by keeping the number of simulation runs to a minimum is a nontrivial problem. The proposed algorithm employs a systematic way to enrich an incomplete database by adaptively filling the gaps between the nondominated points obtained from available data. This thesis is intended for solving multiobjective optimization problems for such black box simulations where the structure of the objective functions is not known or is very complex. To tackle the problem of expensive simulation runs and to reduce the number of true function evaluations, computationally cheap(er) approximations (known as surrogates) of the underlying complex functions are used. Statistical sampling is a crucial part of the surrogate building process which becomes nontrivial in higher dimensional search domains. The novel idea proposed in [18] of using the optimization algorithm DIRECT as a sampling strategy is employed in the current work as well.

A popular approach in the optimization community, hybrid optimization, is where several different optimization techniques are combined to improve robustness and blend distinct strengths of different approaches [24]. This same notion has been extended to multiobjective optimization problems in [49]. The Pareto front approximation method of this thesis is a hybrid approach using two direct search methods, DIRECT (to explore the design space globally) and MADS (to fine tune the potentially optimal regions returned by DIRECT), to balance between global and local searches and to improve the convergence rate.

2.3. The Multiobjective Optimization Algorithm

The proposed new method for multiobjective optimization applies to possibly nonsmooth functions. The ordering property $\big($for biobjective pairs$(f_1(x), f_2(x))\big)$ available in two dimensions is exploited in the biobjective case ([18], [19]) to find the most isolated point from a set of nondominated points at each iteration. However, due to the lack of ordering in higher dimensions, the same strategy cannot be implemented in the multiobjective case. Hence, an alternative using the concept of triangulation, from simplical topology and computational geometry, is proposed in this thesis. Section 2.3.1 describes this alternate approach, and the general scheme of the multiobjective optimization algorithm is presented in Section 2.3.2.

2.3.1. An Alternative Using Delaunay Triangulation

In the biobjective optimization case, a point with the largest Euclidean distance from its neighbors is selected as the most isolated point provided that the nondominated data points are ordered in either of the two objectives. However, this notion of ordering does not work in higher dimensions. Hence, an alternative is proposed for identifying the most isolated point in a high dimensional ($p > 2$) objective space, using the concept of triangulation from simplical topology and computational geometry. Triangulation of a discrete set of points is a subdivision of the convex hull of the points into simplices (e.g., a set of triangles in two dimensions or tetrahedra in three dimensions). A frequently used and studied point set triangulation (in two dimensions) is the Delaunay triangulation, which has the property that no point in the point set falls in the interior of the circumcircle of any triangle in the triangulation. Once the triangulation is constructed for the current nondominated point set, the average distance of each vertex from its neighbors (given two vertices $v^{(1)}$ and $v^{(2)}$, $v^{(1)}$ is a neighbor of $v^{(2)}$ [and $v^{(2)}$ a neighbor of $v^{(1)}$] if there is an edge between $v^{(1)}$ and $v^{(2)}$ in the given triangulation) is computed, and the vertex with the largest average distance from its neighbors is then considered as the most isolated point for that iteration. This triangulation concept generalizes to p-simplices in p dimensions (a triangle is a 2-simplex, a tetrahedron is a 3-simplex, etc.). The precise mathematical formulation of this process is presented in the next subsection.

2.3.2. The Optimization Algorithm

As a preprocessing step, a global search is performed using DIRECT to explore the design space in unsampled regions of a database. If the database is empty, the global exploration step becomes the data generation step. Since DIRECT is an optimization algorithm, the design space exploration is conducted in an intelligent manner by focusing the global search in potentially optimal regions only. For this initialization step, $p+1$ single objective formulations using $p+1$ weight vectors w are obtained — p of which correspond to the individual optima of the p objective functions and one more with equal preference given to all the objectives ($w_i = 1/p$, $i = 1, \ldots, p$, $\sum_{i=1}^{p} w_i = 1$). A peculiarity of DIRECT is that it never samples the boundary points of a design space and takes a large number of iterations to reach reasonably close to the boundary. On account of this behavior of DIRECT, the potentially optimal regions returned by DIRECT are fine tuned using MADS. For each of these $p+1$ single objective formulations using the data collected in DIRECT's global search an interpolating surrogate is built over the entire design space and optimized using MADS. These $p+1$ candidate solutions are evaluated and stored in the database.

The preprocessing step is a crucial part of the proposed algorithm. It helps in eliminating a subset of local Pareto optimal points and accelerates the process of moving to the Pareto front. However, it may not eliminate all the local Pareto optimal points (as the preprocessing is done using a fixed set of weights), and hence the algorithm may not always find the global Pareto front.

A set $X^{(0)}$ of nondominated points (with respect to all database points) is obtained at the end of the preprocessing step that serves as input for the iterative procedure of the algorithm that follows. Each iteration of the algorithm consists of three steps: (1) finding the most isolated point from the current nondominated point set, (2) constructing surrogates for the objective functions with the isolated point determined in Step 1 as the center, and (3) solving several single objective optimization problems to produce candidate Pareto solutions. At the beginning of the iterative process, a trust region radius Δ_0, the trust region contraction parameter ρ, and the tolerance value τ for the trust region radius are initialized.

It is possible to have two nondominated points $x \neq y$ with $F(x) = F(y)$. In this case only one point is used in the algorithm, and the other point(s) are stored as long as x remains nondominated.

Step 1: Isolated point determination

Let $X^{(k)} = \{x^{(k,1)}, \ldots, x^{(k,J_k)}\}$ be the set of nondominated points at the start of the iteration k and $P^{(k)} = \{F(x^{(k,1)}),\ F(x^{(k,2)}),\ \ldots,\ F(x^{(k,J_k)})\}$ be the corresponding objective space set, where $F(x) = (f_1(x),\ \ldots, f_p(x))$ for p objectives and J_k is the cardinality of $X^{(k)}$ (and $P^{(k)}$). Note that for the very first iteration ($k = 0$) the nondominated set $X^{(0)}$ is the result of the preprocessing step. For a typical p-objective optimization problem the Pareto front is a $(p-1)$-dimensional manifold. Hence to construct a $(p-1)$-dimensional triangulation from a set of p-dimensional points, each point $(f_1,\ f_2,\ \ldots,\ f_p)$ is transformed to homogeneous coordinates $(f_1/f_p,\ f_2/f_p,\ \ldots,\ f_{p-1}/f_p,\ 1)$ assuming that the objectives are appropriately scaled. (Precisely shift $f_p(x)$ by an amount s so that $f_p(x) + s > 0$ and $f_i(x)/(f_p(x) + s) = O(1)$ for all i.) Essentially the triangulation is done in the $(p-1)$-dimensional real projective space $\mathbb{P}^{p-1}$. A Delaunay triangulation D_{Tr} is constructed using this $(p-1)$-dimensional transformed data set. The gap $\delta_j^{(k)}$ between the nondominated points in the objective space set $P^{(k)}$ is computed for each point $F(x^{(k,j)})$, $j = 1,\ \ldots,\ J_k$, in $P^{(k)}$ by,

$$\delta_j^{(k)} = \frac{\sum_{i \in N_j} \left\| F(x^{(k,j)}) - F(x^{(k,i)}) \right\|_2}{|N_j|},$$

where $N_j = \{i \mid F(x^{(k,i)}) \text{ is a neighbor of } F(x^{(k,j)}) \text{ in } D_{Tr}\}$. The point $x^{(k,j)}$ (not already accepted as Pareto optimal) with the largest $\delta_j^{(k)}$ value is selected as the most isolated point, and is used as the center point $\tilde{x}^{(k)}$ for a local model with a trust region radius Δ_k. In case of ties, choose the smallest index j. If $N_j = \phi$ ($J_k = 1$), then $\tilde{x}^{(k)}$ is the single point in $X^{(k)}$.

All the most isolated nondominated points $\tilde{x}^{(k)}$ found, together with an associated trust region radius Δ_k, are stored in a database. If $\tilde{x}^{(k)}$ happens to be a previously visited point for $k > 1$, i.e., if for some $i < k$, $\tilde{x}^{(k)} = \tilde{x}^{(i)}$, then the trust region radius is modified as $\Delta_i := \rho\Delta_i$, and if the new Δ_i value is greater than τ, $\tilde{x}^{(k)}$ is the center point for the local model for the next step; otherwise $\tilde{x}^{(k)}$ is accepted as a Pareto optimal point (and excluded from further consideration as an isolated point) and the above procedure is repeated until a new isolated point is found. If $\tilde{x}^{(k)}$ is not a previously visited point, then $\tilde{x}^{(k)}$ and the trust region radius $\Delta_k := \Delta_0$ for the iteration k are added to the database. If the point $\tilde{x}^{(k)}$ is some previously visited point then the algorithm failed to produce any better points (i.e., a point that dominates the center point) at the ith iteration. This behavior could be seen in either of two situations: the surrogate for the local trust region for the ith iteration was not accurate enough or $\tilde{x}^{(k)}$ is a locally Pareto optimal point. Reducing the trust region radius (and generating a new experimental design) would help in constructing a more accurate surrogate to check if any better points around $\tilde{x}^{(k)}$ exist.

In classic trust region algorithms a trust region is expanded when the surrogate is in reasonably good agreement with the true objective function. In the proposed algorithm the center point, and hence the trust region, at each iteration changes as a result of the process of identifying the most isolated point, and weights are redetermined at each iteration that changes the resulting objective function. As a result, the objective function changes at each iteration, and hence is not plausible that a good surrogate for one iteration is likely good for the next iteration as well. Hence, expansion of the trust region is omitted in the proposed algorithm.

Step 2: Surrogates

DIRECT sampling.

The algorithm uses a linear Shepard method (LSHEP from [47]) to approximate a response surface within a trust region. The isolated point $\tilde{x}^{(k)}$ determined in the previous step serves as the center point for the local trust region with radius Δ_k taken as the intersection of the local trust region box $\left\{x \mid \|x - \tilde{x}^{(k)}\|_\infty \leq \Delta_k\right\}$ with the box $[l, u]$ defined by the variable bounds $l \leq x \leq u$. A tuning parameter ϵ for DIRECT controls the exploration mode for DIRECT (local versus global). The number of points to be sampled can be specified as an input to DIRECT, as for a Latin hypercube, the difference being deterministic adaptive rather than random point placement. The points to be sampled using DIRECT are based on the value of the aggregate objective constructed as a convex combination of the p objective functions. $p+1$ different weight vectors, the same as those used for preprocessing are used here by DIRECT to sample the design space. After these $p+1$ runs of DIRECT, a data set $D^{(k)} = \big\{ \big(x, f_1(x), f_2(x), \ldots, f_p(x)\big) \mid x \in X_d^{(k)} \big\}$ of design samples $x \in X_d^{(k)}$ and function values $f_1(x)$, $f_2(x)$ …, $f_p(x)$ is generated.

Surrogate construction.

p LSHEP [47] surrogates $\tilde{f}_1$, $\tilde{f}_2$, …, $\tilde{f}_p$ are built for the p objective functions using the database $D^{(k)}$. At every iteration, several single objective optimization problems are formulated using convex combinations of these surrogates. p of the weight combinations used have fixed values (e.g., (1, 0, … ,0)) for each iteration to compute the individual optima of the p objective functions. Additional weights are derived using an adaptive weighting scheme. Based on the value of the objective functions at the center point $\tilde{x}^{(k)}$ for an iteration and its neighbors, $\max\{1, |N_k|\}$ additional weight vectors w indexed by $N_k = \big\{ i \mid F\big(x^{(k,i)}\big)$ is a neighbor of $F\big(\tilde{x}^{(k)}\big)$ $\big\}$ are generated as follows.

If $J_k = 1$, $w^{(p+1)} = (w_1^{(p+1)}, w_2^{(p+1)}, \ldots, w_p^{(p+1)}) = (1/p, 1/p, \ldots, 1/p)$. Now consider the case $J_k \geq 2$ $(N_k \neq \phi)$. For each $l_i \in N_k = \{l_1, l_2, \ldots, l_{|N_k|}\}$, define a weight vector $w^{(p+i)}$ by

$$w_j^{(p+i)} = \begin{cases} 0, & \text{if } \left|f_j\big(\tilde{x}^{(k)}\big) - f_j\big(x^{(k,l_i)}\big)\right| = 0; \\ \dfrac{c_i}{\left|f_j\big(\tilde{x}^{(k)}\big) - f_j\big(x^{(k,l_i)}\big)\right|}, & \text{otherwise,} \end{cases}$$

where $c_i > 0$ is a normalization constant such that $\sum_{j=1}^{p} w_j^{(p+i)} = 1$. Thus, there are $p + \max\{1, |N_k|\}$ weight vectors $w^{(i)}$ used to construct aggregate objectives for MADS in the next step. All these weight vectors $w^{(i)}$ then define surrogates

$$\sum_{j=1}^{p} w_j^{(i)} \tilde{f}_j(x) \approx \sum_{j=1}^{p} w_j^{(i)} f_j(x)$$

for use in Step 3.

Step 3: Solving optimization problems

Each of the surrogates constructed in Step 2 is optimized using MADS to get a candidate solution. Thus at the end of iteration k, $p + \max\{1, |N_k|\}$ candidate Pareto solutions are obtained, evaluated to get their true function values, and then nondominated points among those are selected. Let $X_*^{(k)}$ be the set of these nondominated points. An updated set $X^{(k+1)}$ of nondominated points is obtained at the end of iteration k by comparing all points from the union $X^{(k)} \cup X_d^{(k)} \cup X_*^{(k)}$. Redundant nondominated points ($x \neq y$ for which $F(x) = F(y)$) are removed from $X^{(k+1)}$ and stored separately. Redundant dominated points are deleted.

These three steps are iterated by the algorithm until a stopping condition is satisfied. In practice, termination is either determined by a fixed number of iterations, or when the value of the star discrepancy (discussed in detail in the next section) drops below a predefined threshold.

2.3.3. Algorithm Pseudocode

Algorithm

Input: p (number of objective functions), v (number of design variables), l, u (bounds on objective functions), Δ (trust region radius), ρ (trust region contraction parameter), τ (tolerance for the trust region radius), N_F (total number of true function evaluations), $\bar{N}_F$ (budget for N_F), D_T (database of all true function evaluations), $D^{(k)}$ (database of true function evaluations within kth local trust region), $w^{(i)}$ (weight vectors), $N_k = \{i \mid F\big(x^{(k,i)}\big)$ is a neighbor of $F\big(\tilde{x}^{(k)}\big)\}$ in a triangulation D_{Tr}

Output: sets $X^{(k)}$ of nondominated points and $P^{(k)}$ of corresponding objective function values.

```
for $i := 1$ step 1 until $p$ do
  begin
    $w^{(i)} :=$ $i$th standard basis vector;
    $w_i^{(p+1)} := 1/p$;
  end
for $i := 1$ step 1 until $p+1$ do
  begin
    minimize $\sum_{j=1}^{p} w_j^{(i)} f_j(x)$ using DIRECT;
    store all points $x$ sampled by DIRECT and
    corresponding objective function values $F(x) = \{f_1(x), \ldots, f_p(x)\}$ in $D_T$;
  end
```

```
Using LSHEP and $D_T$, build $p$ surrogates $\tilde{f}_i(x)$, $i = 1, \ldots, p$ for the $p$ objective functions;
for $i := 1$ step 1 until $p + 1$ do
  begin
    minimize $\sum_{j=1}^{p} w_j^{(i)} \tilde{f}_j(x)$ using MADS;
    Evaluate and store the candidate solutions in $D_T$;
  end
$N_F = |D_T|$;
$k := 0$;
Obtain the sets $X^{(k)}$ and $P^{(k)}$ of nondominated points and
corresponding objective function values with respect to $D_T$;
$J_k := |X^{(k)}|$;
$X^{(k)} = \{x^{(k,1)}, \ldots, x^{(k,J_k)}\}$ and $P^{(k)} = \{F(x^{(k,1)}), \ldots, F(x^{(k,J_k)})\}$;
Remove redundant nondominated points ($x \neq y$ for which $F(x) = F(y)$)
and corresponding objective function values, and
store them separately as $Y^{(k)}$ and $Q^{(k)}$, respectively;
while $N_F < \bar{N}_F$ do
  begin
    Transform each point in $P^{(k)}$ to homogeneous coordinates $(f_1/f_p, \ldots, f_{p-1}/f_p, 1)$;
    Construct Delaunay triangulation $D_{Tr}$ for the transformed dataset;
    for $j := 1$ step 1 until $J_k$ do
      Compute $\delta_j^{(k)}$ using the definition of Step 1;
    $L := \{1, \ldots, J_k\}$;
    $accept := false$;
    while $accept = false$ and $L \neq \phi$ do
      begin
        Find smallest $m$ such that $\delta_m^{(k)} = \max_{j \in L} \delta_j^{(k)}$;
        $\tilde{x}^{(k)} := x^{(k,m)}$;
        if $k > 0$ then
          if $\tilde{x}^{(k)} = \tilde{x}^{(i)}$ for some $i < k$ then
            begin
              $\Delta_i := \rho \Delta_i$;
              if $\Delta_i > \tau$ then
```

```
                $accept := true$;
              else
                $L := L \setminus \{m\}$;
            end
          else
            begin
              $\Delta_k := \Delta_0$;
              Store $\tilde{x}^{(k)}$, $F(\tilde{x}^{(k)})$, and $\Delta_k$;
              $accept := true$
            end
        else
          begin
            Store $\tilde{x}^{(0)}$, $F(\tilde{x}^{(0)})$, and $\Delta_0$;
            $accept := true$
          end
    end $\tilde{x}^{(k)}$ acceptance loop
  if  $L = \phi$ then
    begin
      $X^{(k)} := X^{(k)} \cup Y^{(k)}$;
      $P^{(k)} := P^{(k)} \cup Q^{(k)}$;
      Return $X^{(k)}$ and $P^{(k)}$;
    end
  Build a local trust region $LTR = \big\{x \mid \|x - \tilde{x}^{(k)}\|_\infty \le \Delta_k\big\}$;
  for $i := 1$ step $1$ until $p$ do
    begin
      minimize $\sum_{j=1}^{p} w_j^{(i)} f_j(x)$ using DIRECT;
      Store the point set $X_d^{(k)}$ sampled within LTR by DIRECT
      and corresponding objective function evaluations in $D_T$;
      Let $D^{(k)} = \big\{\big(x,\ f_1(x),\ \ldots,\ f_p(x)\big)\big| x \in X_d^{(k)}\big\}$;
    end
  Using $D^{(k)}$ and LSHEP, build $p$ surrogates $\tilde{f}_j(x)$ for $p$ objectives;
```

```
    Formulate $p$ single objectives $s_i(x) := \sum_{j=1}^{p} w_j^{(i)} \tilde{f}_j(x)$ for $i = 1, \ldots, p$;
    **if** $J_k = 1$ **then**
      $s_{p+1}(x) := \sum_{j=1}^{p} w_j^{(p+1)} \tilde{f}_j(x)$;
    **else**
      **for** $i := 1$ **step** 1 **until** $|N_k|$ **do**
        **begin**
          Compute $w_j^{(p+i)}$ using the definition in Step 2;
          $s_{p+i}(x) := \sum_{j=1}^{p} w_j^{(p+i)} \tilde{f}_j(x)$;
        **end**
    **for** $i := 1$ **step** 1 **until** $p + \max\{1, |N_k|\}$ **do**
      **begin**
        Minimize $s_i$ within the LTR using MADS;
        Evaluate and store the candidate solutions in $D_T$;
      **end**
    $X_*^{(k)}$ := set of nondominated points among the $p + \max\{1, |N_k|\}$ solutions;
    $X^{(k+1)}$ := nondominated points in $X^{(k)} \cup X_d^{(k)} \cup X_*^{(k)}$;
    Remove redundant nondominated points ($x \neq y$ for which $F(x) = F(y)$) and
    corresponding objective function values from
    $X^{(k+1)}$ and $P^{(k+1)}$ and add them to $Y^{(k)}$ and $Q^{(k)}$ giving $Y^{(k+1)}$ and $Q^{(k+1)}$, respectively;
    Remove dominated points from $Y^{(k+1)}$ and $Q^{(k+1)}$;
    $N_F := |D_T|$;
    $k := k + 1$;
    $J_k := |X^{(k)}|$;
  **end** $\bar{N}_F$ while loop
Return ($X^{(k)}$, $P^{(k)}$, $Y^{(k)}$, and $Q^{(k)}$);
```

2.4. Numerical Evaluation

The performance of the proposed method is evaluated using test problems from [17], which are formulated in order to evaluate the capability of a multiobjective optimization algorithm to handle some of the inherent difficulties in MOPs. The main features of the test suite from [17] are the simplicity of problem construction, scalability to any number of design variables and objective functions, a priori knowledge of the Pareto front, and variety of complexities in the Pareto fronts. The test suite tests the ability of an algorithm to accommodate complexity and variety in Pareto front landscapes, converge to the Pareto front, and maintain a good spread of the Pareto optimal solutions. The selection of test problems in this thesis is intended to illustrate the capability of the proposed method to handle diverse landscapes of Pareto fronts, discontinuity in the objective functions, and scalability to higher dimensional design and objective spaces.

2.4.1. Performance Measures

Obtaining good distribution of the nondominated set is a nontrivial aspect of a multiobjective optimization problem in general, and hence dispersion of the nondominated solution set is a good performance measure for comparing different algorithms. There is no known measure in the literature that better assesses the spread of the nondominated set. Most approaches in the literature are based on some statistical measure such as Euclidean distance, standard deviation, or skewness, which may not be sufficient for assessing the distribution of a point set in higher dimensions. Hence, a precise mathematical measure, star discrepancy, was proposed by the authors in [18] for assesing dispersion for a biobjective case. A new technique is proposed in this thesis for extending the star discrepancy based measure to problems with more than two objectives.

General discrepancy [32] of a sequence $S_N = \{s_i^N\}$, where $i = 1, \ldots, N$ and $s_i \in [0,1]^d$ is defined as

$$D(\beta; S_N) = \sup_{B \in \beta} \left| \frac{A(B; S_N)}{N} - \lambda_d(B) \right|,$$

where β denotes a family of Lebesgue measurable subsets of $[0,1]^d$, $A(B;S_N)$ is the number of points from S_N that are also elements of B, and $\lambda_d(B)$ is the d-dimensional Lebesgue measure of B. For practical purposes, a variation of general discrepancy, star discrepancy [12], [32], is often used. Let J^* denote the family of Lebesgue measurable subsets of $[0,1]^d$ of the form $\prod_{i=1}^{d}[0, v_i)$. Then the star discrepancy of the sequence S_N is defined as

$$D_N^*(S_N) = D(J^*; S_N).$$

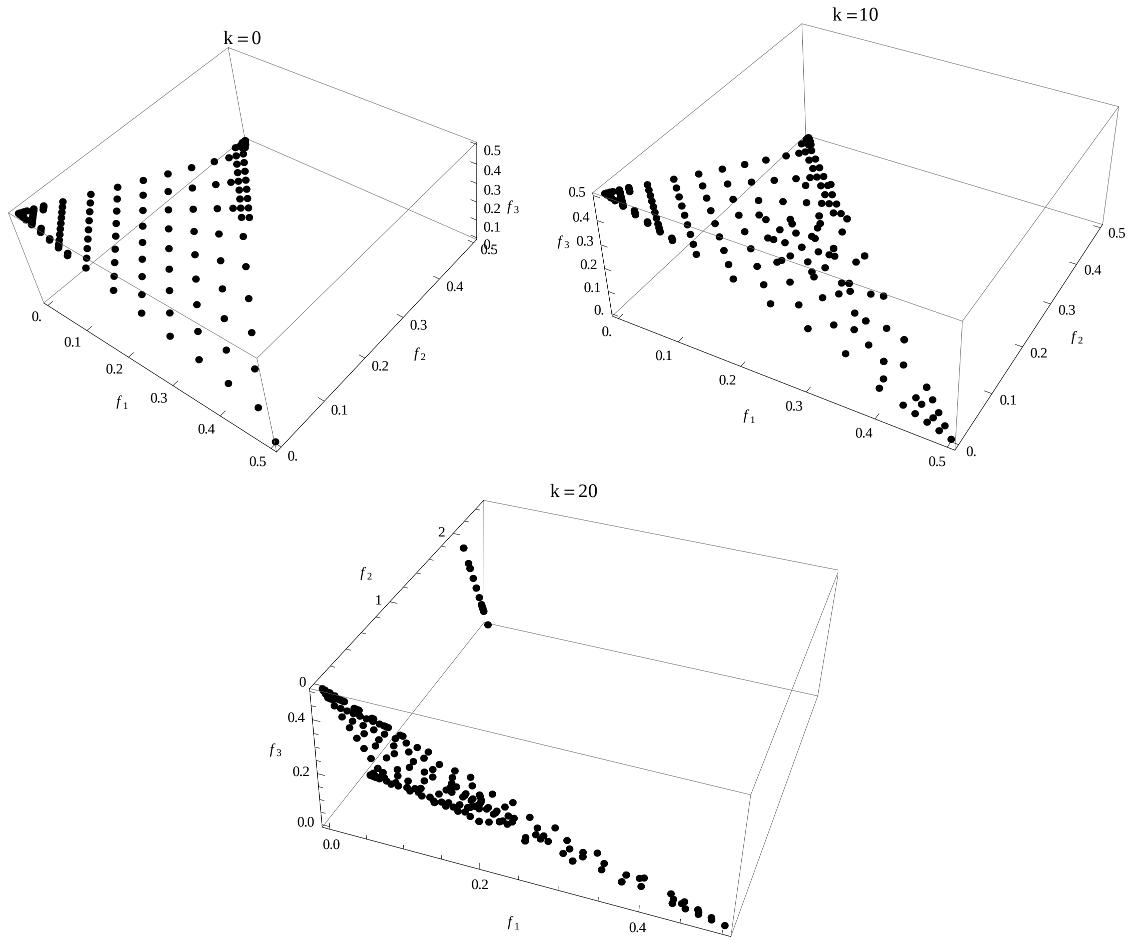


Figure 1. Problem 1 with a planar Pareto front: Pareto fronts after preprocessing, 10 iterations, and 20 iterations of the algorithm (clockwise).

A family of sequences $\{S_N\}_{N=1}^{\infty}$ is said to be uniformly distributed if and only if

$$\lim_{N\to\infty} D_N^*(S_N) = 0.$$

Typically, Pareto fronts in p-objective optimization problems are $(p-1)$-dimensional manifolds, e.g., for a typical triobjective optimization problem the Pareto front would be a two-dimensional surface. The volume of a Pareto front is then approximated by adding up volumes of all the simplices in the corresponding triangulation (e.g., the volume of a triobjective optimization Pareto front is its area, which is approximated by adding the areas of all the triangles in the corresponding triangulation), and then normalized to unit volume. The star discrepancy is computed by approximating J^* with unions of adjacent simplices in the triangulation of the Pareto front.

Another performance measure used to evaluate the proposed algorithm is representativeness (number of nondominated points obtained) N_{ND}. Efficiency in terms of the total number of true function evaluations N_{FE} is another good measure for assessing the performance of a multiobjective optimization method. However, the total number of true function evaluations is set as a budget (and hence a stopping criterion) for all the test problems presented in this thesis, and hence is set to a predefined number. The percentage of nondominated points with respect to the total number of true function evaluations N_{ND}/N_{FE} indicates the ability of a method to eliminate dominated points and focus its effort on obtaining more nondominated points.

The Pareto optimal fronts for the five test problems along with all three performance measures (D_N^*, N_{ND}, and N_{ND}/N_{FE}) along with the number N_{FE} of true function evaluations are presented in Section 4.3.

2.4.2. Parameter Setup

All the test problems presented in Section 2.4.3 have v design variables x_i with bound constraints $0 \leq l_i \leq x_i \leq u_i \leq 1$, and p conflicting objectives f_1, f_2, $\ldots$, f_p to minimize. The numerical evaluation uses a Fortran 95 implementation VTDIRECT95 [26] of the algorithm DIRECT and a C++ implementation NOMAD (version 3.5.1) of the algorithm MADS [5].

The initial parameter setup for VTDIRECT95 for the preprocessing step is (using VTDIRECT95 variable names) design variables lower bound, $L := [l_1,\ \ldots,\ l_v]$, design variables upper bound, $U := [u_1,\ \ldots,\ u_v]$, upper limit on the number of true function evaluations, MAX_EVAL := 2000, and $EPS := 0.001$, where EPS is a tuning parameter that controls the exploration mode of DIRECT (local versus global). VTDIRECT95 is run $p+1$ times with this setup for $p+1$ single objective optimization problems of the form $\sum_{i=1}^{p} w_i f_i(x)$, where the vector w is set to optimize individual objectives for the first p runs and to $(1/p, \ldots, 1/p)$ for the last run. If a point already exists in the database within a tolerable distance (1E−13) from the current sampled point, then the objective function values in the database are used instead of repeating the system analysis at the sampled point. For the preprocessing step, the starting point $x^{(0)}$ for MADS is set to the center point of the entire design space, bound constraints are set to the design space boundaries, and the maximum number of surrogate evaluations is set to 500.

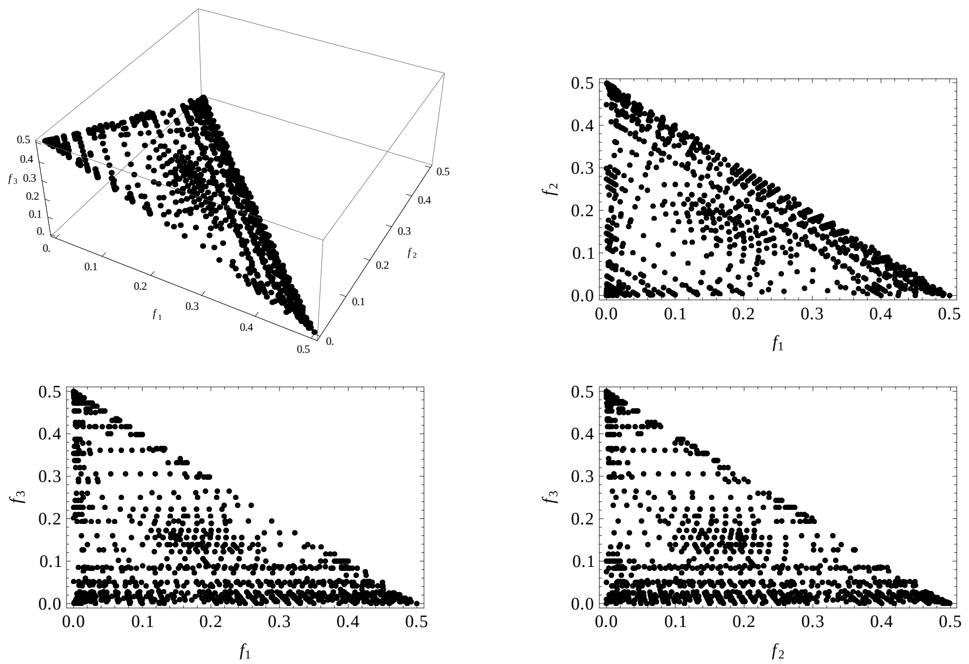


Figure 2. Problem 1: Planar Pareto front using proposed method with 30000 function evaluations.

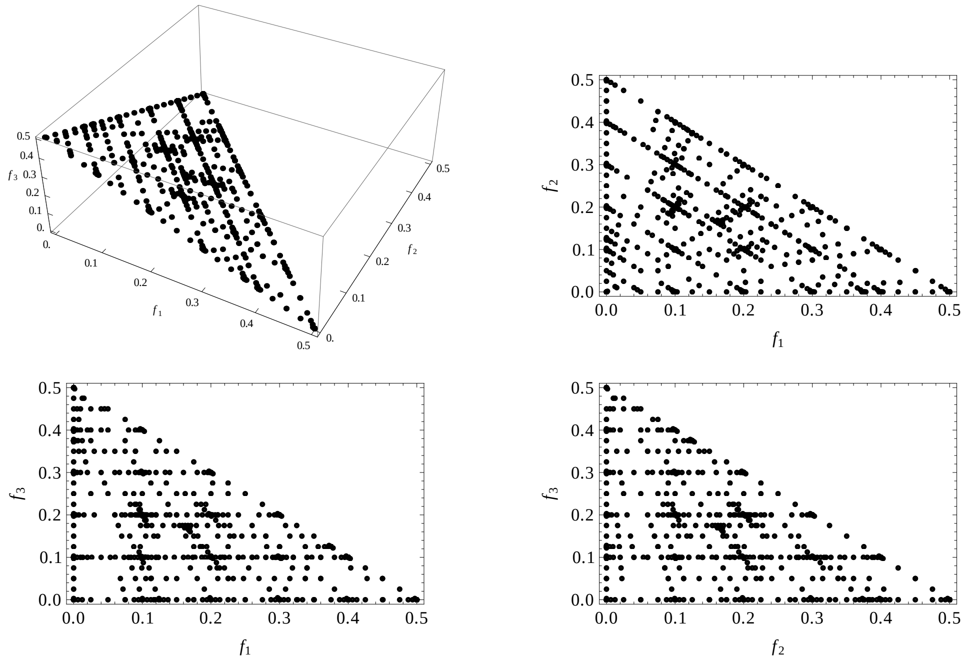


Figure 3. Problem 1: Planar Pareto front using MultiMADS with 30000 function evaluations.

The parameter settings for MADS and VTDIRECT95 for the main iterative procedure are: MADS: the starting point $x^{(0)}$ is set to the center point determined for that iteration, the bound constraints for the MADS run are set to the local trust region boundaries, and the maximum number of surrogate evaluations to 50. VTDIRECT95: L and U are set to the local trust region boundaries for that iteration, MAX_EVAL := 100, and $EPS := 0.1$. The single objective formulation is the same as the one used in the preprocessing step. The local trust region parameters are set as $\Delta := 0.2$, $\rho := 0.5$, and $\tau := 0.02$.

2.4.3. Test Problems and Results

The proposed method is evaluated by five bound constrained problems from [17]. Diversity in the test problems lies in the landscape (a hyperplane for the first problem and a sphere for the second and fourth problems) of the Pareto front, scalability (the fourth test problem is a replica of the second problem with thrice the number of design variables and the last problem has $p = 6$) to higher dimensional design and objective spaces, and discontinuity in the objective functions (the third test problem has a Pareto front with four connected components). For all the test problems, let $x = \big(x_1, \ldots, x_p, \ldots, x_v\big)$. All the test problems except for the last one have three objective functions ($p = 3$); the last problem has six objective functions ($p = 6$). The number v of design variables is set to 12 for the problems 1, 2, and 5, to 20 for the problem 3, and to 36 for the problem 4, all described in subsequent subsections. A budget of thirty thousand true function evaluations (this was suggested by Deb et al. [17] to solve the test problems using evolutionary algorithms) is set as a stopping criterion for all the experiments.

For the first four test problems a figure with four plots with four different views in (f_1, f_2, f_3), (f_1, f_2), (f_1, f_3), and (f_2, f_3) spaces is presented. In addition, for the test problem 1 a figure with three additional plots is presented for illustrating progress of the proposed algorithm at different points in time (preprocessing, after ten iterations of the algorithm, and after twenty iterations of the algorithm). No plots are presented for the problem 5 owing to the difficulty of visualizing higher dimensional objective space. The three performance measures along with the number N_{FE} of true function evaluations for all the test problems are presented in Table 1.

Problem 1. A problem with a planar Pareto front

The problem with the planar Pareto front ([17], Problem $DTLZ1$) is

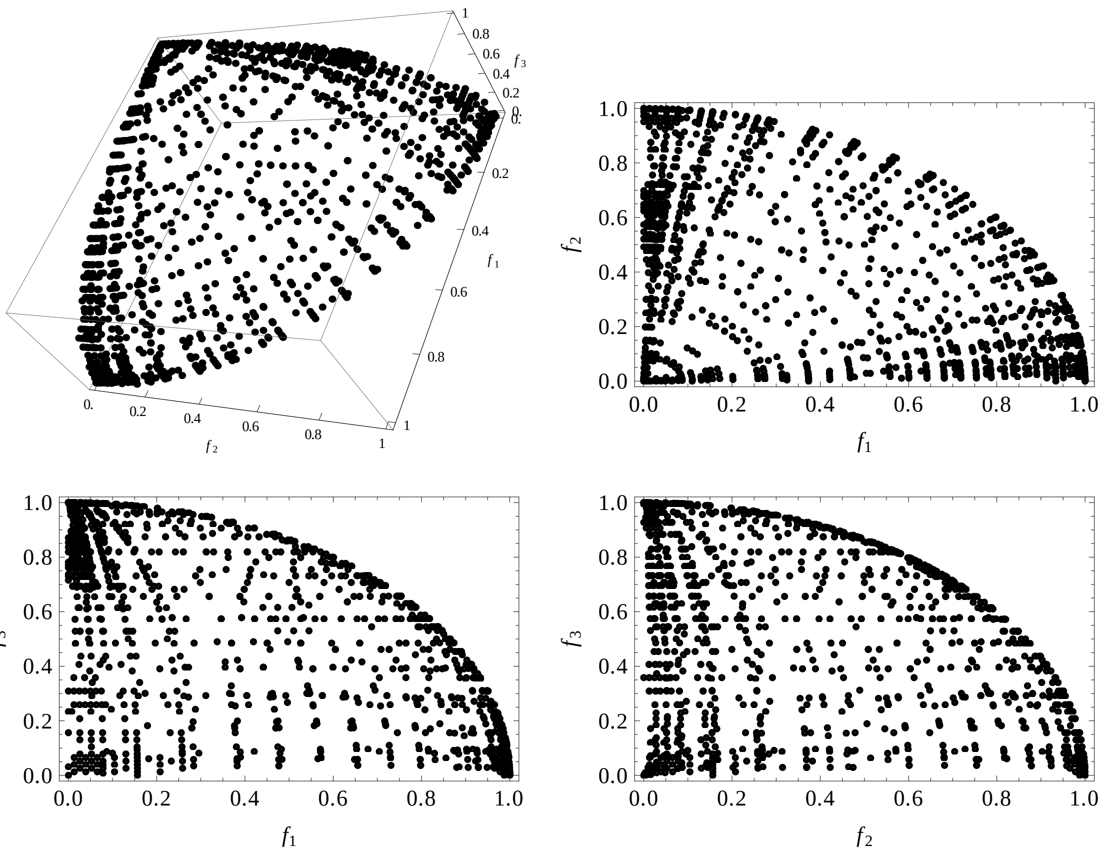


Figure 4. Problem 2: Spherical Pareto front using proposed method with 30000 function evaluations.

$$\min \begin{cases} f_1(x) = 1/2x_1x_2\dots x_{p-1}\big(1+g(x_p,\dots,x_v)\big), \\ f_2(x) = 1/2x_1x_2\dots(1-x_{p-1})\big(1+g(x_p,\dots,x_v)\big), \\ \vdots \\ f_{p-1}(x) = 1/2x_1(1-x_2)\big(1+g(x_p,\dots,x_v)\big), \\ f_p(x) = 1/2(1-x_1)\big(1+g(x_p,\dots,x_v)\big), \end{cases}$$

$$g(x_p,\dots,x_v) = 100[(v-p+1) + \sum_{i=p}^{v}(x_i-0.5)^2 - \cos(20\pi(x_i-0.5))],$$

subject to $0 \le x_i \le 1$, for $i = 1, \dots, 12$. The set of Pareto optimal solutions is $\{x \in \mathbb{R}^{12} : 0 \le x_1 \le 1,\ 0 \le x_2 \le 1,\ x_i = 0.5\ \forall\ i = 3\ ,4,\ \dots,\ 12\}$. The Pareto front is the simplex $\{f_i \ge 0 : \sum_{i=1}^{3} f_i = 0.5\}$. The difficulty in this problem, introduced by the multimodal function g, is to converge to the Pareto front. The search space of the problem contains $11^{10} - 1$ local Pareto fronts. The results are presented in Figures 1 and 2, and Table 1.

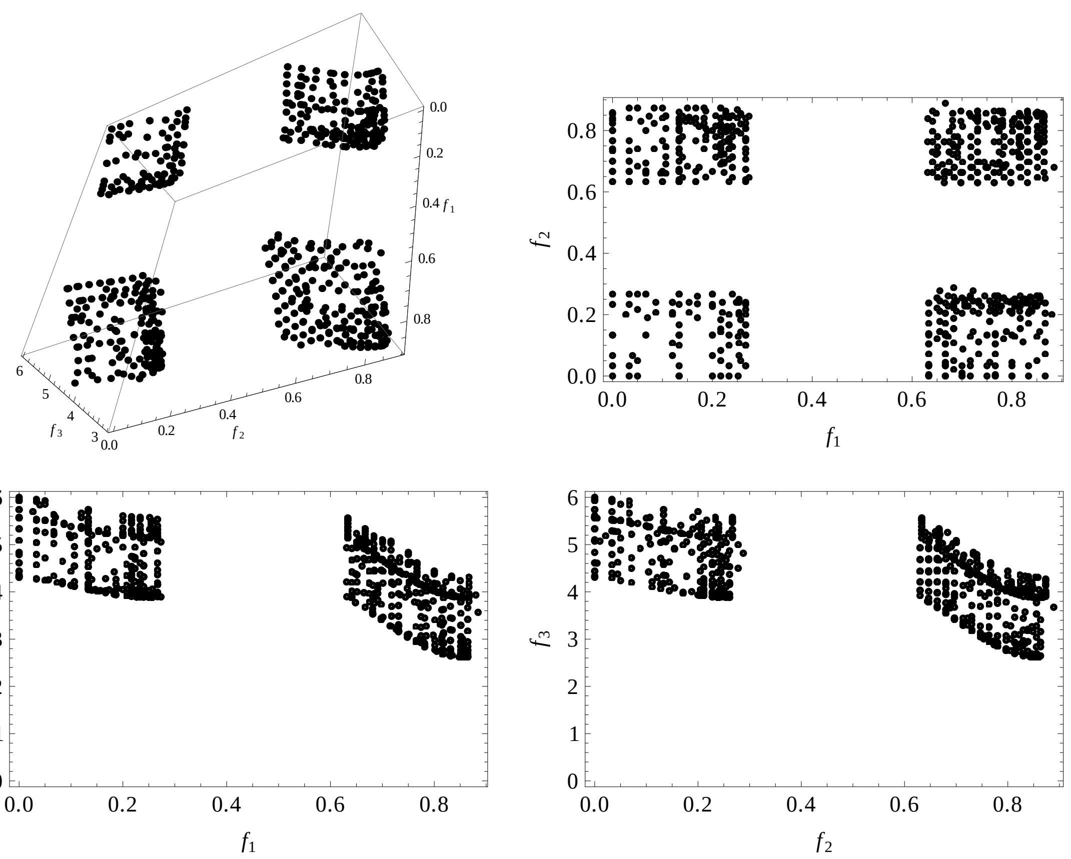


Figure 5. Problem 3: Discontinuous Pareto front using proposed method with 30000 function evaluations.

Results show that the proposed method successfully produces well distributed points near the global Pareto front. The plots in the objective space (Figure 1) and the corresponding performance measures (first three rows in Table 1) at different times (after preprocessing, ten iterations, and twenty iterations of the algorithm) during the course of the algorithm show that the proposed method gradually moves towards the global front by eliminating local Pareto optimal solutions (see the two local Pareto fronts of the third plot in Figure 1). The value for the star discrepancy keeps dropping during the course of the algorithm, which is an indication that the algorithm fills in the gaps around the most isolated point at each iteration as intended. A sudden rise in the star discrepancy value for the third plot (and the third row of Table 1) in Figure 1 is because of the huge gap in the objective space created due to two local fronts. Figure 2 and the fourth row in Table 1 present the progress of the algorithm after exhausting the budget of thirty thousand true function evaluations.

Problem 2. A problem with a Pareto front on a sphere

The problem with the spherical Pareto front ([17], Problem $DTLZ2$) is

$$\min \begin{cases} f_1(x) = \cos(x_1\pi/2)\cos(x_2\pi/2)\ldots\cos(x_{p-2}\pi/2)\cos(x_{p-1}\pi/2)\big(1+g(x_p,\ldots,x_v)\big), \\ f_2(x) = \cos(x_1\pi/2)\cos(x_2\pi/2)\ldots\cos(x_{p-2}\pi/2)\sin(x_{p-1}\pi/2)\big(1+g(x_p,\ldots,x_v)\big), \\ \vdots \\ f_{p-1}(x) = \cos(x_1\pi/2)\sin(x_2\pi/2)\big(1+g(x_p,\ldots,x_v)\big), \\ f_p(x) = \sin(x_1\pi/2)\big(1+g(x_p,\ldots,x_v)\big), \end{cases}$$

$$\text{where } g(x_p,\ldots,x_v) = 100[(v-p+1) + \sum_{i=p}^{v}(x_i-0.5)^2 - \cos(20\pi(x_i-0.5))],$$

subject to $0 \le x_i \le 1$, for $i = 1, \ldots, 12$. The set of Pareto optimal solutions is $\{x \in \mathbb{R}^{12} : 0 \le x_1 \le 1,\ 0 \le x_2 \le 1,\ x_i = 0.5\ \forall\ i = 3, 4, \ldots, 12\}$. The Pareto front is the first octant of the unit sphere $\sum_{i=1}^{3} f_i^2 = 1$. The difficulty in this problem, introduced by the multimodal function g, is again to converge to the Pareto. The search space of the problem contains $3^{10}-1$ local Pareto fronts. Results are presented in Figure 4 and Table 1 (row 5), and show that the algorithm achieves a very good approximation to the global Pareto front maintaining good spread of the nondominated points.

Problem 3. A problem with a discontinuous Pareto front

The problem with a discontinuous Pareto front ([17], Problem $DTLZ7$) is

$$\min \begin{cases} f_1(x) = x_1, \\ f_2(x) = x_2, \\ \vdots \\ f_{p-1}(x) = x_{p-1}, \\ f_p(x) = \big(1+g(x_p,\ldots,x_v)\big)h(f_1, f_2, \ldots, f_{p-1}, g(x_p,\ldots,x_v)), \end{cases}$$

$$g(x_p,\ldots,x_v) = 1 + \frac{9}{v-p+1}\sum_{x_i \in X_P} x_i,$$

$$h(f_1, f_2, \ldots, f_{p-1}, g(x_p,\ldots,x_v)) = p - \sum_{i=1}^{p-1}\Big[\frac{f_i}{1+g(x_p,\ldots,x_v)}(1+\sin(3\pi f_i))\Big],$$

where $X_P = \{x_p, \ldots, x_v\}$, $0 \le x_i \le 1$, for $i = 1, \ldots, 20$ and $p = 3$. This test problem has $2^{p-1} = 4$ connected components. The set of Pareto optimal solutions is $\forall x_i \in X_P,\ x_i = 0$. The difficulty in this test problem is to identify the connected regions in the objective space.

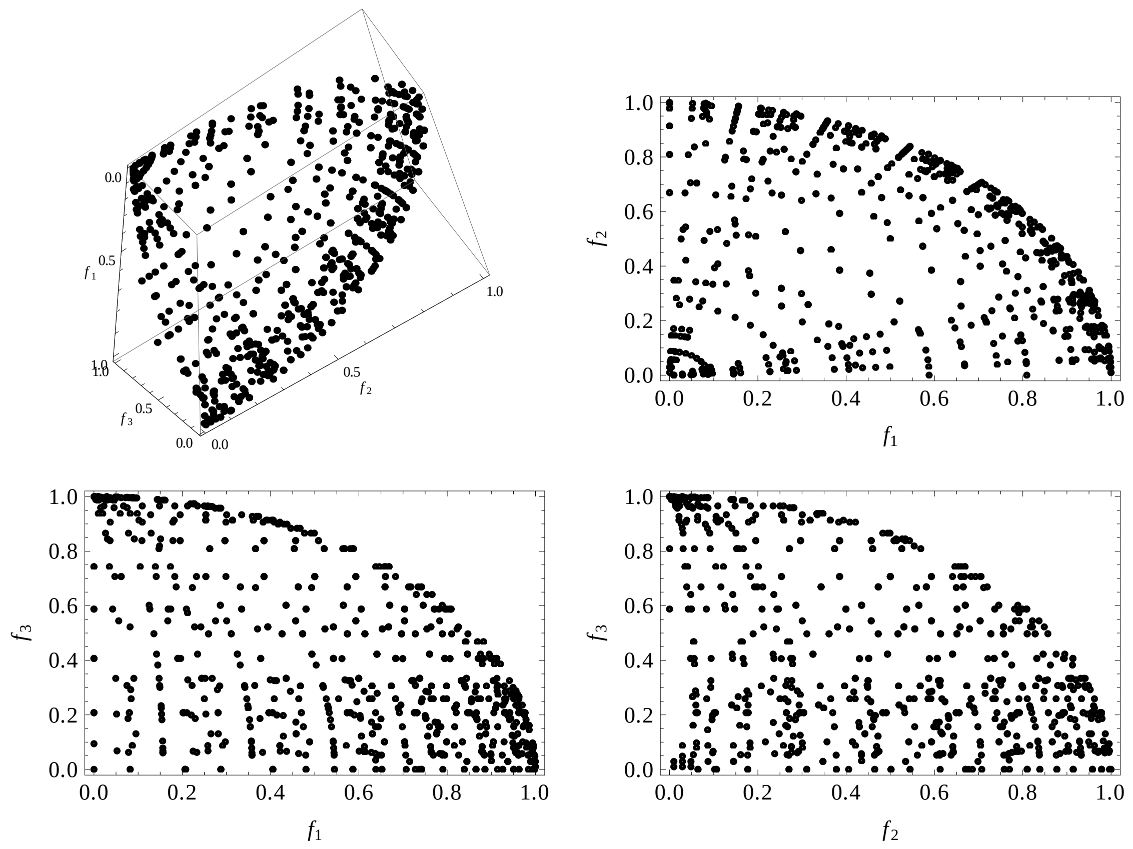


Figure 6. Problem 4: Pareto front in higher dimensional design space using proposed method with 30000 function evaluations.

Results obtained by applying the proposed method are illustrated in Figure 5 with four different plot views of the objective space, and Table 1 (row 6) presents the performance measures for this test problem after thirty thousand true function evaluations. Results show that the method successfully identifies the four connected components by maintaining a good spread of nondominated points in each region. Computing the star discrepancy for a discontinuous Pareto front is rather difficult and has not been considered further here.

Problem 4. A problem with a high dimensional search space

This problem is a replica of the test problem 2 with the only difference being the number v of design variables set to 36. The intention was to test the ability of the proposed method to handle a higher dimensional design space. Results (Figure 6 and Table 1 (row 7)) show that the method successfully handles a higher dimensional design space and the results are comparable to those obtained with a lower dimensional design space for test problem 2.

Problem 5. A problem with a 6-dimensional objective space

This problem ([17], Problem $DTLZ3$) uses the objective function definitions of the problem 4.3.2 except for a different function g and the number of objective functions $p = 6$. Here the function g is

$$g(x_p, \dots, x_v) = \sum_{x_i \in X_P} (x_i - 0.5)^2,$$

where $X_P = \{x_p, \dots, x_v\}$, subject to $0 \le x_i \le 1$, for $i = 1, \dots, 12$. The performance measures are presented in Table 1 (last row). Results show that the method successfully produces a good collection (N_{ND}) of evenly spread (D^*) Pareto optimal solutions for a higher dimensional objective space.

2.4.4. Comparison with other Algorithms

As mentioned in the introduction, the proposed method falls under the classical scalarization approaches for solving multiobjective optimization problems. Hence it is logical to compare the proposed method with other deterministic multiobjective optimization methods rather than with the evolutionary approaches. The method is compared to state-of-the-art techniques in the class of deterministic approaches for solving MOPs: BiMADS ([6]) for the biobjective case and MultiMADS ([7]) for the multiobjective (more than two objectives) case. Results for three test problems (a biobjective problem from 2.4.4 using BiMADS, the triobjective Problem 1 and the six-objective Problem 5 from 2.4.3 using MultiMADS) are presented and compared to those obtained using the proposed method.

Comparison with BiMADS on a biobjective optimization problem

The biobjective optimization problem, an engineering design problem from [4], with two objectives f_1 (fundamental vibration frequency) and f_2 (weight), and two design variables x_1 (height) and x_2 (base) is

$$f_1(x) = x_1 x_2,$$

$$f_2(x) = -\left(\frac{1/\left(\frac{1}{K_s} + \frac{L^3}{3E x_1 x_2^3/12}\right)}{W/g} \right)^{1/2},$$

where $x = (x_1,\ x_2)$, $0.5 \le x_1 \le 1$, $0.2 \le x_2 \le 2$, and $K_s = 10$, $L = 12$, $E = 3.00E + 07$, $W = 50$, and $g = 386.4$ are the model constants. The C++ implementation from the package

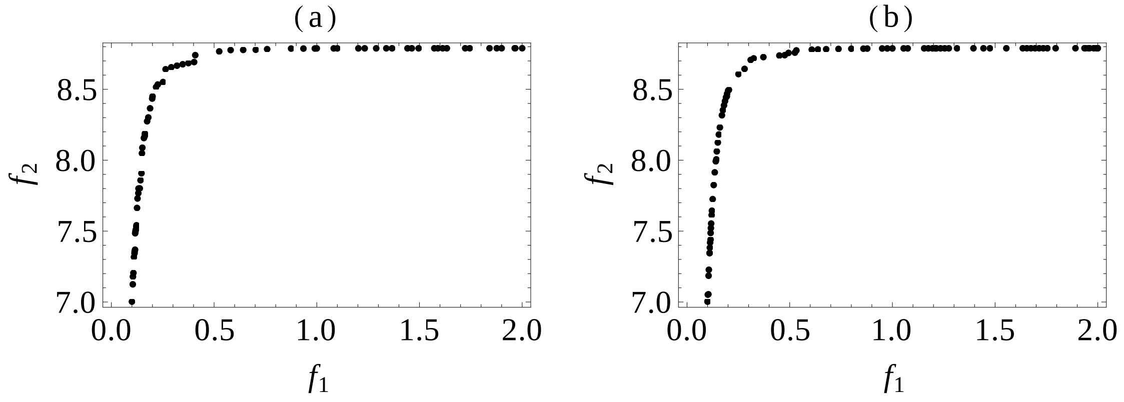


Figure 7. (a) the proposed method and (b) BiMADS with a budget of 200 (maximum) evaluations each for the biobjective optimization problem in 2.4.5

NOMAD 3.5.1 [21] for BiMADS is used with the default settings for all the parameters, and the optimization algorithm presented in [18] is used for handling the biobjective case of the proposed method with the same settings of the tuning parameters that were used in [18]. A budget of two hundred true function evaluations is set for both the methods. Figure 7 shows two plots in the objective space (f_1, f_2) for the two methods, the proposed method and BiMADS, respectively. The performance measures for the two methods presented in Tables 1 and 2 (last rows) show that the nondominated point set distribution for the proposed method compares favorably to that for BiMADS, although the cardinality of the nondominated set produced by BiMADS is better than that using the proposed method. Readers are referred to [18] and [19] for a detailed comparison of the biobjective version of the algorithm with BiMADS on diverse test problems.

Comparison with MultiMADS on Multiobjective Optimization Problems

Results for two of the five test problems, Problem 1 and 5, presented in the previous section are used for the comparison with the results obtained using the multiobjective optimization algorithm MultiMADS [7]. All the default settings in the C++ implementation NOMAD 3.5.1 of the algorithm MADS [5] are used. As discussed in [7] a fifty point Latin hypercube experimental design is used in the initialization of the algorithm MultiMADS to diversify the starting point selection. A budget of thirty thousand true function evaluations is set for both problems with a budget of one thousand evaluations per MADS run except for the construction of the tangent hull where the budget for a MADS run is set to two thousand evaluations. For the triobjective optimization Problem 1 from 2.4.3, MultiMADS calls MADS 29 times, including four calls during the initialization step: three times to compute the shadow minimum and once to construct the tangent hull, and 25 single objective optimizations in the main iterative procedure; for the six objective optimization Problem 5 from 2.4.3, MADS is called 29 times again, including seven calls during the initialization step: six times to compute the shadow minimum and once to construct the tangent hull, and 22 calls to solve 22 different single objective formulations. Both problems are solved using the Euclidean norm based distance formulation for the single objective optimization. Visualizations for the triobjective case are presented in Figure 3. Table 2 shows the performance measures for both problems.

Results are again very favorable for the proposed method in the star discrepancy measure. MultiMADS (and BiMADS for the biobjective case) in general produced a very good collection of nondominated points but the distribution of the point set compared to that using the proposed method is not so good. This is an indication of either larger gaps or clusters of nondominated points in the objective space.

2.4.5. Discussion

Figures 1–2 and 4–6 and Table 1 show that the algorithm successfully finds a fairly good approximation to the global Pareto front with well distributed (D_N^* values in Table 1) Pareto optimal points for diverse types of multiobjective optimization problems. The new adaptive weighting scheme and the guided sampling using DIRECT effectively approximate the Pareto front by gradually moving towards the global front by filling the gaps in the incumbent nondominated point set. In that sense the method gradually learns the shape of the true Pareto front and concentrates its computational effort in potentially optimal regions. Both the quality and the quantity of the obtained nondominated point set improves with the number of iterations of the algorithm (for example, the plots in Figure 1 and first three rows in Table 1). The division of the global (preprocessing) and the local search in the course of the algorithm plays an important role in directing the results towards global Pareto fronts. Although all the local Pareto optimal solutions were not eliminated in the preprocessing step for the problem 1 (see the third plot of Figure 1), those were eliminated

Table 1

Performance measures: proposed method

	D_N^*	N_{ND}	N_{FE}	N_{ND}/N_{FE}
Problem 1 (preprocessing)	0.3202	165	4218	0.0391
Problem 1 (10 iterations)	0.1814	225	5751	0.0391
Problem 1 (20 iterations)	0.5508	258	7261	0.0355
Problem 1 (all)	0.0690	1229	30000	0.041
Problem 2	0.1050	1541	30000	0.0514
Problem 3	—	800	30000	0.0266
Problem 4	0.0403	920	30000	0.0307
Problem 5	0.1198	2409	30000	0.0803
BiObjective	0.0410	71	200	0.35

eventually during the course of the algorithm (see Figure 2). However, this might not always be the case, and the algorithm might find only a local Pareto front. The star discrepancy based measure for assessing the distribution of the obtained nondominated set seems to be working reasonably well, and can be used as a stopping criterion.

The method produced reasonably good results for a multiobjective optimization Problem 5 (2.4.3) with six objective functions. Delaunay triangulation used in this work has a well known [2] worst case bound of $O(n^{\lceil d/2 \rceil})$ for a set of n randomly distributed discrete points in d-dimensional space. Testing the proposed method on problems with a large number of objective functions might divulge useful information about the ability of the algorithm to handle such problems. The comparative study from Section 2.4.4 reveals that the adaptive weighting scheme based on the triangulation technique is very effective in producing well-distributed fronts as compared to other state-of-the-art methods for solving MOPs.

All the test problems presented in this thesis have known analytical forms for the objective functions, which makes it easy to assess the results obtained using the proposed method, and hence provides a way of evaluating the method. This is not the only class of problems that the proposed method targets. As mentioned in the introduction, the method intended for blackbox multiobjective optimization problems with possibly nonsmooth functions where the analytical form of the objectives is not known. The author plans to test the proposed

Table 2

Performance measures: BiMADS and MultiMADS

	D_N^*	N_{ND}	N_{FE}	N_{ND}/N_{FE}
Problem 1 (all)	0.1594	1223	30000	0.0408
Problem 5	0.2335	4938	30000	0.1646
BiObjective	0.1187	89	200	0.46

method on real engineering applications where such blackbox simulation based multiobjective optimization is prevalent.

2.5. Conclusion

A new method is proposed for multiobjective optimization of possibly nonsmooth functions targeting the class of blackbox simulation based optimization problems. The algorithm proposes a new adaptive weighting scheme using the concept of triangulation from simplical topology and computational geometry. Cheap(er) surrogates are employed for the expensive objective functions, and an optimization based guided sampling is used as an experimental design for each local trust region optimization. The method combines two direct search algorithms DIRECT (to explore the design space globally) and MADS (to fine tune the potentially optimal regions returned by DIRECT), to balance between global and local searches and to improve the convergence rate. The method is evaluated using five test problems from the literature with different Pareto front landscapes. Results show that the proposed method achieves a reasonably good approximation to the global Pareto front with a good spread of the nondominated points for all the test problems. Star discrepancy is introduced as a new quantitative measure of point distribution. The method compares very favorably to two other deterministic multiobjective optimization methods in those quantitative measures. In the future, the authors plan to apply the proposed method to real engineering problems where the analytical form of the objectives is not known and the objective function evaluation is very expensive. Another extension of the method would be to consider problems with constraints.

Chapter 3.

ADML: AIRCRAFT DESIGN MARKUP LANGUAGE

3.1. Introduction

Aircraft design by nature is a multidisciplinary process where several different disciplines (see Figure 8) such as geometry, structures, aerodynamics, controls, propulsion, flight mechanics, and so on contribute to achieving an optimal design adhering to all the design constraints for all the disciplines involved. The first step in the design process, the conceptual design, is characterized by a large number of design alternatives and trade-off studies, and a continuous, evolutionary change to the aircraft concepts under consideration [40]. Conceptual design is primarily a search process that requires an extensive exploration of the design space in order to gain insight into the relations between the design variables and the aircraft performance. It formulates a set of design variable quantities, which, according to appropriate modeling principles and design constraints, defines a vehicle that fulfills a set of minimum requirements determined by the vehicle mission. Traditional conceptual design was conducted as isolated disciplines with low fidelity interdisciplinary coupling and mostly linear interactions between disciplines. However, due to rapidly developing computer technology and algorithmic improvements, conceptual design methods have advanced tremendously in the past few decades. Today's modern aerospace systems exhibit strong interdisciplinary coupling and require a multidisciplinary, collaborative approach [42]. Aircraft design has become a collaborative endeavor that involves many individuals from diverse groups around the world working together in an extended enterprise environment to achieve a common goal. Advances in computer capacity and speed, along with increasing demands on the efficiency of the aircraft design process, have intensified the use of simulation based design and analysis tools to explore design alternatives both at the component and system levels. Analysis methods that were once considered feasible only for advanced and detailed design are now available and even practical at the conceptual design stage. Rapid analysis methods also allow multidisciplinary design optimization methods to be implemented in conceptual design. This changing philosophy of conducting the conceptual aircraft design and analysis poses additional challenges beyond those encountered in a low fidelity design and analysis of aircraft. Although the use of sophisticated design and analysis tools has become prevalent in the aerospace community, the field of interdisciplinary communication still remains in a primitive state. Aircraft design systems are not yet equipped with the state-of-the-art in data representation and communication that are prevalent in several other domains. An objective of this thesis is to propose a unified data approach for bridging the gap in data communication methods and making the aircraft design process more agile.

Aircraft design and analysis involves manipulation of a large amount of interdisciplinary data including, but not limited to, inputs and outputs. Efficient transfer, sharing, and manipulation of aircraft design and analysis data across different platforms, applications, and users demands a formal structured representation of data in a well organized data sharing and validation environment. In general, due to the lack of a uniform representation, the same information is duplicated several times, each time in a different format specific to the underlying implementation. In order to exchange this information between different disciplines/applications/users, a translator needs to be designed that converts one format to/from another at every facility and for each format. Thus, the lack of a standard, uniform representation results in redundancy in codes and duplication of information and efforts incurring a lot of maintenance overhead. Another common problem with this kind of data exchange is data inconsistency. All these factors inhibit sharing and exchanging of interdisciplinary data and greatly hinder the conceptual design process making it less efficient. To alleviate this burden, a unified system is sought that provides certain capabilities for modeling the massive amount of multidisciplinary data, such as portability, maintainability, reusability, platform independence, integrity, (syntactic) correctness, and system recovery. With a platform and language independent data exchange standard like XML (extensible markup language), information can flow seamlessly in a heterogeneous environment with diverse computing platforms, programming languages, and hardware systems. This thesis

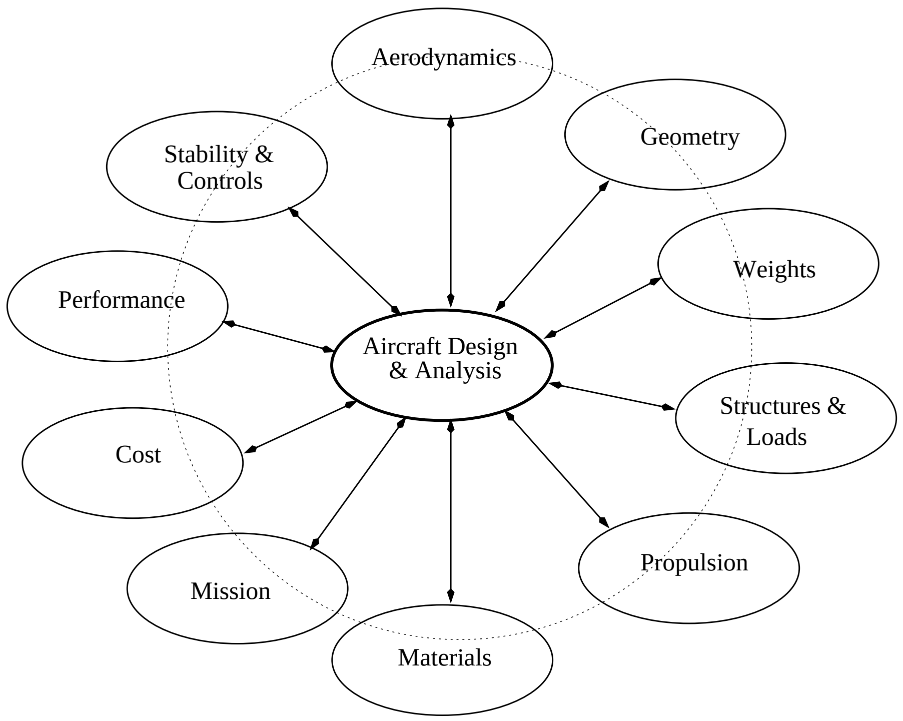


Figure 8. Disciplines involved in an aircraft design process.

proposes some first steps for the conceptual aircraft design and analysis community to move in this same direction.

3.2. Background

3.2.1. Motivation

The former multidisciplinary optimization branch (MDOB) at the NASA Langley research center (LaRC) [34] identified frameworks for multidisciplinary analysis and optimization research, and promoted a multidisciplinary, collaborative approach and sharing of information among disciplines for aircraft design and analysis systems. The advanced engineering environments (AEE) study committee sponsored by NASA has investigated a number of technical, managerial, cultural, and educational barriers that need to be overcome in order to realize a multidisciplinary, collaborative approach [42]. Several design requirements related to information management and integration of tools, systems, and data need to be addressed first in order to realize a unified system. Based on the knowledge gained from the frameworks proposed by the MDOB ([31] and [44]), this section outlines

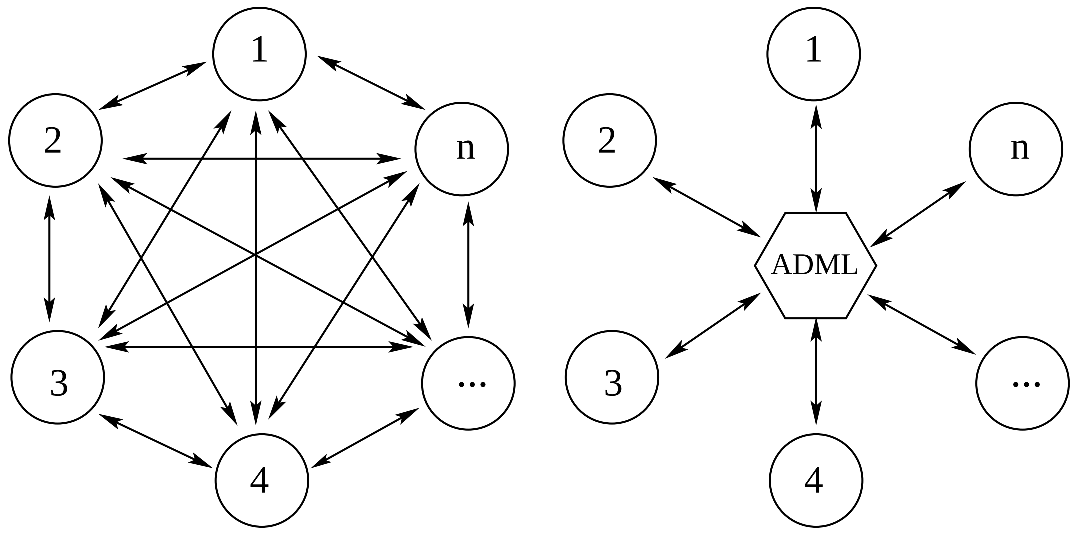


Figure 9. n^2 interfaces (left) without a standard data format vs. n interfaces using ADML (right).

several desirable features related to data management pertinent to multidisciplinary aircraft design, analysis, and optimization.

- Use of standards: Use of standard data formats facilitates maintainability of codes and eliminates duplication of information and effort.
- Data sharing: Intra- and interdisciplinary data sharing in a multidisciplinary, collaborative environment is a crucial feature for solving interoperability issues and automating design processes.
- Extensibility: Advances in aircraft conceptual design processes entail a flexible and extensible data format for supporting different design variants as well as new configurations.
- Platform independence: A platform, language, and vendor neutral format is sought for seamless data communication across different platforms, applications, and users in a multidisciplinary environment.
- Object oriented programming: The data format should support object oriented design principles that facilitate aircraft design processes with several useful capabilities such as data binding and integration, object encapsulation, extensibility through inheritance, flexibility through polymorphism, and so on.

The implementation of a data standard adhering to all these requirements is a major challenge. A platform and language independent format to represent aircraft design and analysis data is a desirable way to meet all these requirements and support a multidisciplinary system distributed across a network of heterogeneous computing environments. This is the motivation behind proposing an XML based data format as a first step towards meeting that challenge.

A multidisciplinary collaborative design environment enables engineers to cooperate by means of structured and mostly autonomous exchange of information. This exchange is mostly conducted through input/output interfaces between design and analysis modules. Hence, the number of interfaces is a critical factor for an efficient exchange of data and a central information model is a key feature. As indicated in Figure 9 (left) the number of interfaces required without a standard data format grows quadratically ($O(n^2)$) with the number n of disciplines, application codes, or users. However, using a unified data format such as XML, the number of interfaces grows linearly in n resulting in ($O(n)$) interfaces.

3.2.2. Existing Data Formats, Standards for Representing Product Data

Several data modeling languages and technologies have emerged over the past two decades or so for representation and exchange of product manufacturing information. IGES ([29], [36]) is a language neutral data format that allows exchange of product data among computer aided design (CAD) systems. A vendor neutral system CAPRIS is described in [25] for accessing a variety of CAD systems though a unified and simple programming interface. CAPRIS maintains a boundary representation (BRep) data structure common to all participating CAD systems. CAPRIS uses SOAP (simple object access protocol) for exchanging structured information and relies on XML for messaging. However, the geometry schema for CAPRIS is not publicly available. A successor of IGES, STEP (standard for exchange of product model data), is a family of standards defining a robust and time-tested methodology for describing product data throughout the lifecycle of a product. STEP is a comprehensive ISO standard (ISO 10303) ([38], [39]) that describes a mechanism to represent and exchange product data and has been widely used in the aerospace, automobile, electrical, electronic, and other industries [10]. As discussed in [37], STEP has a proven record of success in modeling aircraft geometry. The part STEP AP209 (application protocol: composite and metallic structural analysis and related design — ISO 10303-209:2001) of STEP has been developed to address data exchange in a design/analysis/manufacturing process. The second edition of AP209 has recently been renamed as "multidisciplinary analysis and design", and is in the final stage of development as of June 2013 [1]. STEP uses a data modeling language called EXPRESS ([38], [46]) to describe and exchange product data between CAD, CAM (computer aided manufacturing), CAE (computer aided engineering), and other CA* systems. EXPRESS combines ideas from the entity-attribute-relationship family of modeling languages with object modeling concepts. However, unlike XML, EXPRESS is not easily extensible and is not supported by many widely used software tools. Although EXPRESS provides rich facilities for data modeling at the semantic level, unfamiliarity of today's application programmers with the traditional STEP based data modeling techniques impedes its widespread usage. Furthermore, XML has become a de facto standard for representing and exchanging digital data for several domains, including domains that are within the scope of STEP. Since STEP can semantically model the high fidelity information required by many XML applications, the STEP data modeling standard and XML are complementary technologies. It is a logical next step to merge the traditional STEP technology within XML. With the integration of the two, the best of both worlds can be achieved.

3.2.3. Rationale for using XML

XML, a W3C (World-Wide Web Consortium) recommendation [51], is a standard concomitant with a number of powerful capabilities (extensibility, flexibility, reusability, maintainability, and so on) and a generic, robust syntax for developing specialized markup languages. Unlike HTML (hypertext markup language), XML by itself specifies neither preconceived semantics nor a predefined tag set; it instead provides a means for defining content and semantics of XML documents. One of the major requirements in a multidisciplinary collaborative environment is the data sharing ability to overcome disciplinary isolation. The platform, language, and vendor independent format of XML makes it well-suited to the task of satisfying multidisciplinary aircraft data requirements.

XML is a profile of an existing ISO standard, ISO 8879, known as SGML (standard generalized markup language) [28], and is an acceptable candidate within other ISO standards without further standardization ([34]). The simple ASCII text format of XML allows aircraft applications running on heterogeneous systems with diverse platforms to readily communicate with each other. Aircraft design application written in any programming language can process the same XML document without any modification, thus eliminating redundancy and offering reusability. In addition, the inherent hierarchical nature of XML provides a way to define structural relationships that exist in the data and facilitates application of object oriented principles to conceptual aircraft design data. Name, attributes, and content of an XML element are closely related to class name, properties, and composition associations in an object oriented aircraft design. Thus, with the use of an XML based markup language, it is possible to faithfully model aircraft design and analysis data as well as structural and functional relationships among different data elements.

A variety of XML parsers for almost all high level programming (and scripting) languages are abundantly available for automatic generation and parsing of XML content. XML itself is a metalanguage—a language that is used to define an unlimited number of special purpose markup languages. XML data semantics (grammar) can be specified using either a document type definition (DTD) [51] or an XML schema [50]. An added benefit of using DTD or XML schema is that they provide support for data validation. A data file encoded in XML is considered valid if it complies with the corresponding DTD or XML schema. Without using a schema for an XML document, a separate validation tool needs to be implemented. An XML schema provides additional significant advantages over a DTD, such as more advanced data types and a very elaborate content model. The aircraft design markup language proposed here is based on XML schema.

3.2.4. XML based Markup Languages Pertinent to Multidisciplinary Aircraft Design

There are several XML based languages developed for various application domains. There are compelling examples of success from various disciplines, e.g., a systems biology markup language (SBML) [27] developed for systems biology models and data; MathML [45], an XML based language developed for mathematical notations; Office Open XML, a Microsoft file format (commercial application) for storage of electronic data, and many more.

Although the aerospace industry is no exception for developing XML based standards for exchanging aircraft data and models, there are only a handful of successful examples. The JSBSim flight dynamics model software library [9] is a batch simulation application aimed at modeling flight dynamics and control for aircraft. JSBSim is an XML based model description specification where input files are supplied in XML format. These XML files contain descriptions of aerospace vehicles, engines, scripts, etc. DAVE-ML is an XML based markup language for a draft AIAA flight dynamic model exchange standard [3], inspired by JSBSim, for the interchange of flight dynamics modeling data between facilities. Both JSBSim and DAVE-ML are intended to provide a platform and language neutral format for exchanging flight dynamics modeling, verification, and documentation data where the major XML elements are mathematical objects. However, JSBSim provides its own XML tags for representing mathematical constructs (e.g., product, sum, quotient, etc.), whereas DAVE-ML uses the verbose MathML format for representing mathematical constructs.

An XML based markup language, MatML [8], developed in coordination with the National Institute of Standards and Technology (NIST) targets multiple industries for facilitating the exchange of a wide variety of material properties. The latest version of MatML (MatML3.0 and beyond) ported the language specification from DTD to XML Schema, and has many refinements over previous versions.

The finite element modeling markup language (femML) [13] was proposed to address the data interpretation and application interoperability in the finite element modeling domain. The project was initiated by members of the composite materials and structures group at the Naval Research Laboratory and the International Science and Technology Outreach Society. femML uses MatML as a namespace in its specification.

All these XML based markup languages discussed heretofore target a single or a subset of disciplines involved in a multidisciplinary environment. A recent development effort at the German aerospace center DLR involves a new data exchange format CPACS (common parametric aircraft configuration schema) for representing all the necessary data required for conceptual and preliminary aircraft design and analysis. After evaluating how well the proposed ADML effort fits within the context of CPACS, it was found that the goals of the collaborative exercise at DLR are closely aligned with ADML objectives; however, the fundamental difference is that the ADML effort started bottom-up with powerful constructs for functions and abstract mathematical objects, and with unconventional aircraft configurations in mind, whereas the current version of CPACS started top-down from entire aircraft to single data objects (point lists), and can only currently handle traditional aircraft designs.

3.3. Aircraft Design Markup Language (ADML)

The ADML project started with an intention to address the data communication needs of the recently founded (2009) Collaborative Center for Multidisciplinary Sciences (CCMS) for the development of future aerospace vehicles, involving Virginia Tech, Wright State University, and Air Force Research Laboratory at Wright Patterson Air Force Base (WPAFB), Ohio. The collaborative center specifically investigates multidisciplinary analysis and design of several futuristic aircraft such as the joined-wing SensorCraft, flapping micro air vehicles, and efficient supersonic air vehicles. A flexible, extensible, and comprehensive XML based format ADML is proposed to handle these futuristic aircraft designs.

ADML is based on XML technologies making it human readable and computer processable. It is designed to accommodate data for numerous disciplines involved in the conceptual design phase and can be extended to high fidelity analysis. ADML includes capabilities for a model to be self-validating and self-documenting, with the provenance of a model's components included within the model and transferred with it (see Section 3.3.2(B) for a detailed description).

A specialized grammar of ADML, the ADML schema, provides a format for the exchange of the aircraft design and analysis data, therefore each discipline is required to design import/export tools that comply with the schema one time only. In this data-centric setup the number of interfaces is minimal and effective communication can be established, resulting in substantially reduced cost and time required to exchange aircraft data. Use cases (presented in Section 3.4) have indicated significant reduction in effort to exchange simple models when utilizing this format. Even greater benefits could be attained for large complicated models or more disciplines.

Although an XML based markup language is well-suited for addressing interoperability issues involved in a multidisciplinary, collaborative environment, the actual development is not as easy as it first appears. Developing a generic, comprehensive, and compact XML schema for each and every discipline involved in the aircraft conceptual design phase is a very challenging task. Every discipline has its own set of modeling requirements and constraints that adds up to the overall complexity of the final design. Accommodating new aircraft configurations for futuristic air vehicles is even more challenging, and demands a comprehensive and extensible data format. The inherent hierarchical nature and extensibility of an XML schema plays a significant role in structuring various components of conceptual

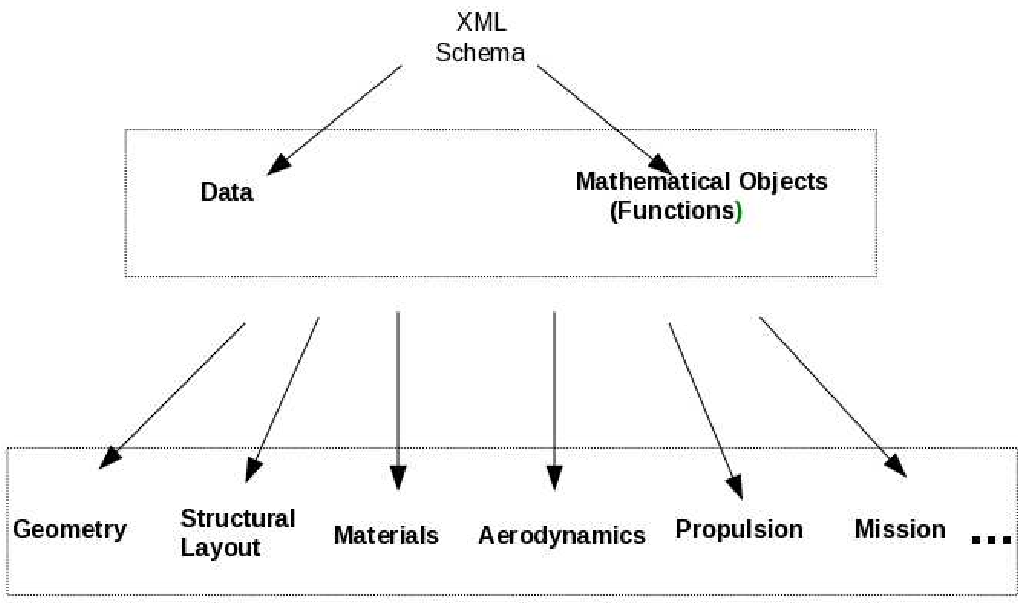


Figure 10. ADML Development Approach.

aircraft design. Figure 10 presents a simplified version of the bottom up development approach of the ADML schema. The specification for the ADML schema would need to include the capability to define aircraft data specific to each and every discipline and component or subsystem of the aircraft involved in the conceptual design phase. An overview of the ADML schema modules (data, functions, basic geometry, and high level aircraft design constructs) and the existing and the future modeling capabilities of the proposed XML schema follow.

3.3.1. Low Level Schemata: Common Components

A) Data Schema

At a very high level, everything is data. However, the rationale for dividing the XML schema in different sections (data, functions, geometry, etc.) is to exploit the functional and logical distinction among different aircraft model objects and to maintain their inherent hierarchy. The data schema is at the lowest level of the hierarchy in a top down view, representing the simplest form of data. Elements of the data schema are used as the building blocks for all other higher level elements.

Most of the elements in ADML require a *name*, a *description*, and a *unit* associated with them. Therefore, a complex type XML element *nd* is defined to encapsulate these elements. All the elements in the data schema, as well as in other ADML modules, that require any or all of these descriptive identifiers (name, description, and/or unit), can be derived from the *nd* element as a base.

The major element of the data schema is the *variable* element. Variables are used to define inputs and/or outputs to/from a design or an analysis. A variable element has a human readable *name*, a *description*, a *unit*, a *value*, a *min*, a *max*, some *flags*, and other *scalar* parameters associated with it, and a machine readable variable identifier, *vid*. A variable defined in an ADML document can be referenced at a later point in a mathematical expression. The value of a variable element consists of a *scalar* (an atomic value) or a *tensor* (a multidimensional array). A tensor is an ADML element defined recursively to represent an array of arbitrary dimensions. A higher rank tensor is defined in terms of a lower rank tensor. A vector (*vtype* element) is a rank 1 tensor and a matrix is a rank 2 tensor. In general, a k-dimensional array can be defined as a rank k tensor, e.g., a $2 \times 3 \times 4$ tensor is defined as a sequence of two 3×4 matrices that are defined in terms of three 4-dimensional vectors each. A typical use of a tensor element could be to define relational data (function tables). An example of a $2 \times 3 \times 4$ tensor follows, omitting the schemata defining the tags.

```
<variable>
  <name>tvar1</name>
  <description>2x3x4 tensor example</description>
  <value>
    <t>
      <t>
        <v>1 2 3 4</v>
        <v>1 2 3 4</v>
        <v>1 2 3 4</v>
      </t>
      <t>
        <v>5 6 7 8</v>
        <v>5 6 7 8</v>
        <v>5 6 7 8</v>
      </t>
    </t>
  </value>
</variable>
```

B) Representing Mathematics

A significant part of aircraft design and analysis data comprises mathematical objects such as functions, expressions, arbitrary dimensional lists, and operators. Therefore, communicating mathematical objects among different entities (applications, users, and/or platforms) plays a crucial role in exchanging data in a multidisciplinary, collaborative conceptual aircraft design and analysis environment. Careful thought has been given to a format for representing mathematical constructs while developing the proposed ADML schema. Three possible candidates are the XML based markup language MathML and two widely used computational software tools, Mathematica and Matlab. The most significant advantage of using a MathML format to represent mathematics is that MathML itself is an XML based markup language and can be parsed and validated easily using available XML parsers; however, MathML is an extremely verbose and unreadable format. Editing mathematical expressions in MathML requires a special editor because the markup is very complex. This makes it impractical to edit by hand. Furthermore, the conceptual aircraft design and analysis community is more interested in communicating content rather than representing mathematical objects. Moreover, Matlab and Mathematica are among the most popular tools used to evaluate mathematical expressions in the aerospace community. Therefore, this thesis proposes the use of Mathematica or Matlab syntax over the verbose MathML format for representing mathematics. However, if an application needs to parse the mathematical data being exchanged at the other end, then parsing subroutines need to be written specific to the underlying implementation. The supported format is more useful when the mathematical objects being exchanged are meant to be passed to either Matlab or Mathematica tools for evaluation. For sharing mathematical data (mathematical lists or arrays) that are meant to be parsed, an application should make use of the more relevant and easy to parse XML element, tensor, defined in the data schema.

The major schema elements for representing mathematical objects include *operator*, *relation*, *mlist*, and *expression*. These elements can be represented in either Mathematica or Matlab format using the *format* attribute associated with them.

An *operator* is a generalization of the familiar notion of a function. Typically, an operator is used to represent the operations performed on functions to produce other functions. An example of an operator on functions, composition with a Bessel function, represented in the Mathematica format follows, omitting the schemata defining the tags.

```
<operator format="Mathematica">
  <name> f </name>
  <description>
    Composition with Bessel function of the first kind, order 0
  </description>
  <domain> AnalyticFunctions </domain>
  <range> AnalyticFunctions </range>
  <arguments> z </arguments>
  <definition>
    f[z_][x_] := BesselJ[0,z[x]]
  </definition>
</operator>
```

Another type of element is the *relation* element. A relation might be defined by an expression that involves logical or relational operations. A relation can also be viewed as a subset of the Cartesian product of k sets. Thus, the first $k - 1$ values in a k-tuple correspond to the arguments or inputs to the relation, and the kth value corresponds to the output. The corresponding relation table can be defined using the *mlist* element. Although an mlist element somewhat resembles a tensor element from the data schema, its intended usage is quite different. A tensor element is primarily used to transfer a multidimensional array across different systems (platforms or users) and not for manipulating the array. However, the intended use of an mlist element is to define and manipulate an arbitrary list structure (where every list element can have a different cardinality). A tensor, being a recursive XML element, facilitates an easy parsing process at the other end, whereas an mlist element has the advantage of a compact representation using either Matlab or Mathematica format. The rationale for having two different elements (expression and relation) to represent mathematical relations is that a relation is a special type of an expression involving only relational operations. The intended use of an expression element is to represent intermediate computations or evaluations in an analysis or a design process.

The schema definition for an expression element and an example of an expression that estimates the drag divergence Mach number (M_{dd}) as a function of an airfoil technology factor (K), the thickness-to-chord ratio (t/c), the lift coefficient (c_l), and the sweep angle (L) follow, again omitting some of the schemata.

```
<xs:element name="expression">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="nd">
        <xs:sequence>
          <xs:element ref="variables" />
          <xs:element ref="definition" />
          <xs:element name="patternStr" type="xs:string" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute ref="format"/>
        <xs:attribute name="eid" type="xs:ID"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

```
</xs:element>
```

```
<expression format=Mathematica>
  <name>M_dd</name>
  <description>
    estimation of drag divergence Mach number
  </description>
  <variables>K, L, t, c, c_l</variables>
  <definition>
    M_dd=K/Cos[L]-(t/c)/Cos^2[L]-c_l/10*Cos^3[L]
  </definition>
</expression>
```

A simple example of an *mlist* (generalization of tensor), omitting some of the schemata, is

```
<xs:complexType name="mlist">
  <xs:complexContent>
    <xs:extension base="nd">
      <xs:sequence>
        <xs:element ref="variables"/>
        <xs:element ref="definition"/>
      </xs:sequence>
      <xs:attribute name="lid" type="xs:ID"/>
      <xs:attribute ref="format"/>
      <xs:attribute ref="structure"/>
      <xs:attribute name="dimension" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```
<mlist format="Mathematica" structure="general">
  <name>L</name>
  <description>an arbitrary list structure</description>
  <definition>
    {{0,0.1 },{{18,-0.1 },{19,-0.09 }},
    {{20,-0.08 },{22,-0.05 },{23,-0.05 }},
    {{25,-0.07 },{27,-0.15 },{90,-0.6 }}}
  </definition>
</mlist>
```

C) Basic Geometry

The ADML schema for aircraft geometry starts with low level, common geometry elements such as *point*, *pointList*, *line*, *plane*, *nurbs*, and *frame*, and builds up more complex components of an aircraft such as airfoils, wings, fuselages, etc. A *point* is defined as a list of real values (coordinates); a *line* is defined using two *point*s; and a *plane* is defined using a *point* and a *normal*.

Another fundamental geometry element is *nurbs*. The geometry schema presented in this thesis supports NURBS (nonuniform rational B-spline) based geometry model to represent curves and surfaces. Each *nurbs* element is defined in terms of a set of control points (the *controlPoints* element), a knot vector (the *knotVector* element), a weight vector (the *weightVector* element), and the NURBS order (the *order* element). The XML schema definition for a NURBS element and an example of a NURBS curve of order three with five control points associated with five weights and eight knots follow.

```
<xs:simpleType name="nurbsType">
   <xs:restriction base="xs:string">
      <xs:enumeration value="curve"/>
      <xs:enumeration value="surface"/>
      <xs:enumeration value="BezierCurve"/>
      <xs:enumeration value="BezierSurface"/>
   </xs:restriction>
</xs:simpleType>

<xs:complexType name="nurbs">
   <xs:complexContent>
      <xs:extension base="nd">
         <xs:sequence>
            <xs:element name="controlPoints" type="mlist"/>
            <xs:element name="knotVector" type="mlist"/>
            <xs:element name="weightVector" type="mlist"/>
            <xs:element name="order" type="xs:integer"/>
         </xs:sequence>
         <xs:attribute name="ntype" type="nurbsType" />
         <xs:attribute name="ncp" type="xs:integer" />
         <xs:attribute name="nurbsID" type="xs:ID"/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>

<nurbs ntype="curve" nurbsID="NC1">
   <name>ncurve1</name>
   <description> NURBS curve of order 3 </description>
   <controlPoints format="Mathematica" structure="array">
      <definition>
         {{0,0,0.5 }, {0,-0.5,0.5 }, {0,-0.5,0 }, {0,-0.5,-0.5 }, {0,0,-0.5 }}
      </definition>
   </controlPoints>
   <knotVector format="Mathematica">
      <definition> {0,0,0,0.5,0.5,1,1,1 }</definition>
   </knotVector>
   <weightVector format="Mathematica">
      <definition> {1,0.707107,1,0.707107,1 }</definition>
   </weightVector>
   <order>3</order>
</nurb>
```

3.3.2. High Level Aircraft Design Constructs

A) Modeling Aircraft Geometry

Rapid development of computer technology over the past decade has changed the conduct of conceptual aircraft design. Aircraft analysis methods that were considered feasible only for advanced and detailed designs are now available and even practical at an early stage of the aircraft design process. To fully exploit the available computing resources and analysis methods, the geometric model of aircraft must be generated rapidly and easily so as not to inhibit the conceptual aircraft design process. However, aircraft geometry is one of the most complex constructs among various conceptual aircraft design components, and likewise the representation of the geometry model is complex.

In the development of any new aircraft, the outer mold line (OML) is key to designers in almost every discipline. Core differences in the utilization of the aircraft geometry often lead to the development of multiple aircraft representations, that cater to different design disciplines, only later to be merged into one final aircraft design. Not only is this process inefficient, it is also difficult to implement in an MDO framework that executes autonomously. Multidisciplinary analysis and design requires a single parametric geometry representation for a configuration that is shared amongst the various disciplines involved. The software behind most commonly used CAD systems is extensive and tailored to serve its community of mechanical engineers. In contrast, computational design optimization is an extension of conceptual design merged with high fidelity computational models; the geometric requirements are significantly specialized compared to general industrial CAD systems. Taking this requirement into consideration, a geometry model supported in ADML is the practical implementation VT-CST ([35]) of class shape transformation (CST, [33]) developed at Virginia Tech. VT-CST is capable of rendering tailless supersonic configurations with embedded engines as well as conventional and joined-wing configurations. In addition, ADML can handle other types of parametric geometries such as boundary representation (BRep) constructs using NURBS curves.

B) Representing an Entire Aircraft

As mentioned previously, ADML development follows a bottom-up approach where all the basic common components are defined first, and other more complex high level constructs are built using the low level schemata as and when required. The root elements for the low level schemata are *data*, *functions*, and *geometry*, and that for the high level aircraft design and analysis is *aircraft.* Each *aircraft* element consists of one or more instances of *model* and *analyses* elements. An aircraft design is described using a *model* element, which consists of *wings*, *fuselages*, *landingGears*, and *propulsion* as subelements along with some catalog elements such as *materials*, *performance*, *mission*, and *global*. Figure 11 shows the hierarchy for the first few levels of the ADML schema. Owing to the complexity of the ADML schema and the large number of XML elements needed to represent aircraft design components, it is not feasible to list and discuss each and every element in this thesis. Instead a brief discussion of the high level elements follows with the full ADML schema in Appendix A.

< airfoils >

The *airfoils* element consists of a sequence of *airfoil* elements. Each *airfoil* element is defined, using the choice data structure available in the XML Schema, as a choice among a parametric definition, a NURBS based cross section, a VT-CST based geometry definition, or a string reference to an external definition (for example a NACA airfoil definition). An airfoil defined in such a way can be referenced in a *wing* definition using the associated *uID* attribute.

< pointMasses >

A primary goal in airplane conceptual design is to determine an estimate of the mass and the moment of inertia tensor of the airplane. There are many internal components of an airplane that are difficult and/or unnecessary to precisely define until a later stage in the design cycle. However, the mass and the moment of inertia tensor of these objects must still be accounted for in the conceptual airplane design as they have a nonnegligible mass thus directly impacting the structural design and performance of the airplane. Therefore, instead of creating a separate definition for each individual type of object, only a simple description of the location of the component, its mass, and its moment of inertia tensor are required. Usually the mass and the moment of inertia tensor of such components are estimated using empirically based methods and a simple Cartesian location is used to place the object in or on the airplane. In the ADML schema, a *pointMasses* element has been created to account for any component of the airplane which could be described in this way. Each *pointMasses* element is defined as a sequence of *pointMass* elements. A *pointMass* is defined using four elements: *mass*, *inertiaTensor*, *location*, and *provenance*.

< structuralElements >

The primary members that make up an aircraft structure are beams and plates. However, in structural analysis using the finite element method, beams can be and are often modeled as plates depending on their shape. In modern aircraft design, one may wish to evaluate a structure made up of multiple different materials including both metals and laminated fiber composites. For example, one design may utilize laminated fiber composite spars whereas another uses traditional aluminum spars. In order to handle this variation in material properties (and thus the number of design variables necessary), two descriptions of the properties of a plate were developed, one for isotropic materials (metals) and one for laminated composites (fiber composites and sandwich panels). This definition is generic enough that it can be used to define the properties of all of the primary structural members of a wing: the spars, the ribs, and the skin panels.

The *structuralElements* element consists of a sequence of two elements, *structElementIsotropic* and *structElementComposite*, corresponding to plates made up of isotropic and composite elements, respectively. A section of a wing can have vectors of references (using the associated *uID* attribute) to these elements. This in effect divorces the material properties and the thicknesses of the structural members from the structural layout allowing for a designer to easily switch material properties and thicknesses by simply changing the reference numbers to the properties.

< wings >

The *wings* element consists of a sequence of *wing* elements that define instances of wings and/or tails of an aircraft. Each *wing* element is defined as a sequence of *planform*, *structure*, and *controlEffectors* elements, and a set of attributes. The airfoil geometry can be in VT-CST format or NURBS based. In ADML, a wing *structure* is defined as a set of *section*s; each *section* is defined in terms of ribs and spars (using number of, thickness, materials, etc.). This simplification is assumed to be sufficient for a wing definition in the conceptual design phase. Each *section* can have a different number (thickness and material) of spars and ribs, that way adding flexibility to accommodate a myriad of wing configurations.

< fuselages >

A *fuselages* element consists of one or more *fuselage* elements that follows VT-CST's parametric geometry definition from [35] drawn from a cross section class function (in the Y-Z plane) defined along a distribution class function (in the X-Z plane), both of which are scaled with the length and width of the desired fuselage.

< landingGears >

A *landingGears* element is comprised of a sequence of one or more *landingGear* elements chosen from three different configurations, tricycle, quadricycle, and multibogey, each defined using a set of design parameters (mass, lowered and raised coordinate combinations in X, Y, Z, etc.). A tricycle or a multibogey configuration has one nose gear centered on the aircraft body whereas a quadricycle configuration has two nose gears.

< propulsion >

Each *propulsion* element consists of four subelements, *engines*, *cowls*, *ramps*, and *EEWSs* (EEWS stands for engine exhaust washed structures), and follows VT-CST's parametric definition. As mentioned earlier, VT-CST was developed to design tailless supersonic aircrafts with embedded engines. An *engines* element is defined as a sequence of one or more *engine* elements that are defined as a set of design parameters. Likewise each of these *cowls*, *ramps*, and *EEWSs* elements is defined as a sequence of one or more *cowl*, *ramp*, and *EEWS* elements, respectively.

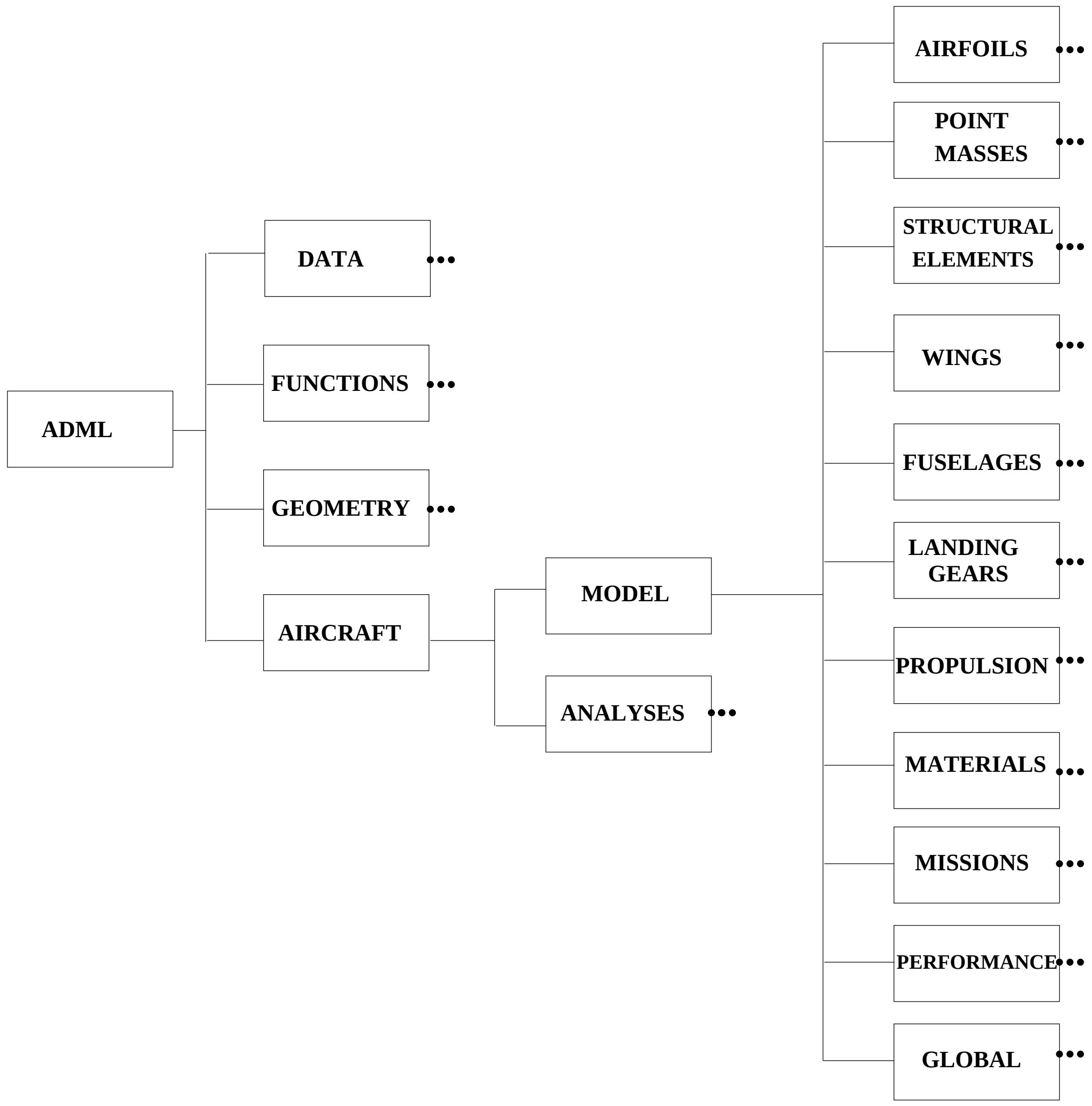


Figure 11. ADML taxonomy.

< materials >

The data defined for the *materials* element is classified as a reusable dataset, and a reference to material IDs is provided in other elements to encode their material properties. Two types of materials—isotropic and orthotropic—are defined in a *materials* element.

< missions >

Here, a list of *missions* can be specified. The *missions* are built up from *mission* segments, which allow for simple conceptual design definitions. An aircraft uses a reference to one or more of these *missions* as its design missions. A mission segment is a specific maneuver that the airplane is designed to perform. For example, a mission segment for the aircraft to cruise would contain the desired cruising altitude, cruising Mach number, and a specified distance or flight time. The missions are typically used in a flight performance analysis and optimization. The *missions* element is again classified as a catalog element, and is defined outside the aircraft *model* element.

< **performance** >

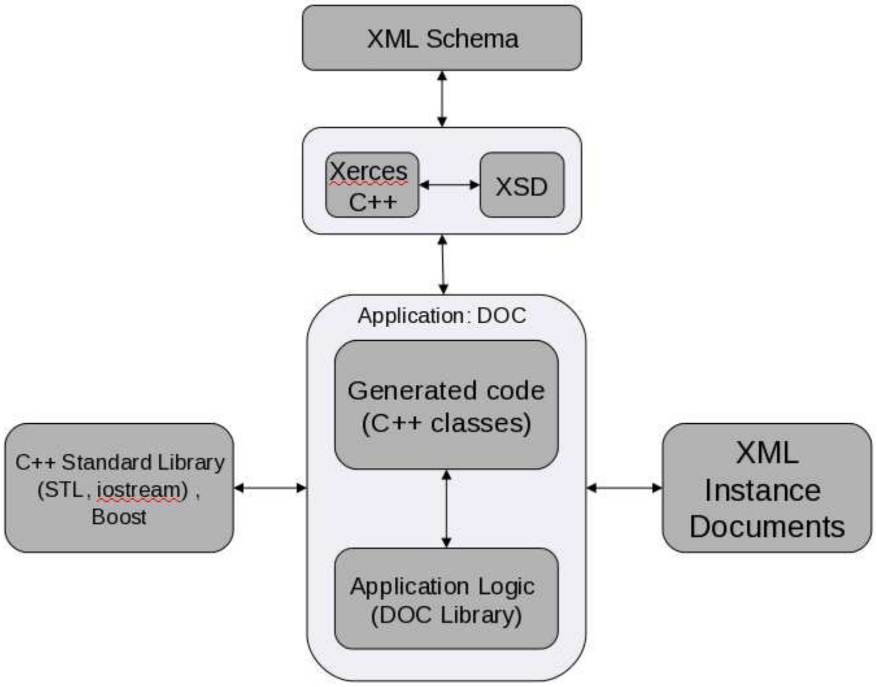
A set of performance parameters (maximum cruise Mach number, maximum altitude, maximum range, dive speed, maximum load factor, wing loading, etc.) is defined in this high level node. Some of the performance parameters are inputs and others are consequences of analyses.

All these high level constructs constitute the third level in the ADML taxonomy as shown in Figure 11.

3.3.3. Features of ADML

Modular development

Modular schema development facilitates logical decomposition of XML elements into subsets where each individual subset focuses on specific functional capabilities thereby enabling reusability. Each small subset or module that results from this exercise can work as a building block for other more complex modules thereby enabling extensibility. The inherent modular or hierarchical structure of multidisciplinary aircraft design elicits modular schema development. The top level modules in the XML taxonomy, data and mathematical objects, serve as the foundation for developing more complex aircraft design constructs that appear at a lower level in the inheritance hierarchy. Every discipline involved in an aircraft design phase can be viewed as a separate module in the XML schema development process and can be used either as a single, isolated entity or as a part of a hierarchical structure built by combining several disciplines together.



XML Schema
Xerces C++
XSD
Application: DOC
Generated code (C++ classes)
Application Logic (DOC Library)
C++ Standard Library (STL, iostream) , Boost
XML Instance Documents

Figure 12. Integration of geometry schema with DOC project.

Object Oriented Approach

A W3C XML schema, with a hierarchical type system, closely resembles an object oriented programming paradigm. Amongst the significant features of an XML schema are extensions (and restrictions), element references, and an object like behavior of an element (that carries attributes and other elements). The modular schema development of the proposed schema, as discussed in the previous subsection, facilitates reusability and extensibility. All the high level elements (corresponding to high level constructs in an aircraft design) in ADML follow an object oriented programming approach. The aircraft design applications such as VT-CST (written in C++) that use object oriented technologies can greatly benefit from this by converting the ADML schema to the classes of the high level language, and then accessing the schema elements as objects of those classes.

Provenance Capability

A provenance capability is provided for all the high level constructs to describe the origin or history of the associated data, and is defined as an XML string describing author, date, etc.

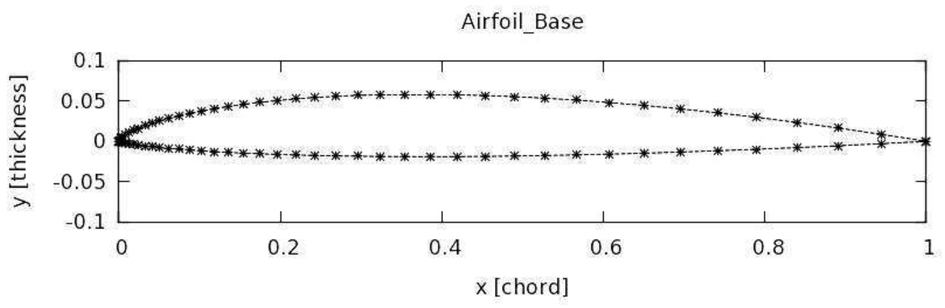


Figure 13. Airfoil geometry.

3.4. Use Cases

3.4.1. Airfoil Shape Optimization

A C++ project, design optimization in C++ (DOC) [11], that computes design sensitivities for conceptual aircraft design applications is used as a pilot project to demonstrate an application of the proposed ADML schema. An open source, cross platform W3C schema to C++ data binding compiler, Codesynthesis XSD, is used to convert the XML schema to C++ classes. Once the C++ classes are generated from the XML schema, the data stored in XML instance documents can be accessed through the C++ objects (member variables and functions) rather than dealing with the intricacies of reading and writing XML. The software architecture of the application in Figure 12 depicts the geometry integration and related tools/packages. The XSD software uses Xerces-C++ as the underlying XML parser. Xerces-C++ is a validating XML parser written in a portable subset of C++ and is available under the Apache Software License.

The geometry schema presented in this thesis supports a NURBS (nonuniform rational B-spline) based geometry model to represent curves and surfaces, as for the airfoil shown in Figure 12. Below is the XML schema definition corresponding to the C++ code for a NURBS structure. Each *nurbs* element is defined in terms of a set of control points (the *controlPoints* element), a knot vector (the *knotVector* element), a weight vector (the *weightVector* element), and the NURBS order (the *order* element).

A sample code listing for the ADML schema definition for a NURBS based airfoil object follows.

```
<xs:element name="airfoilType">
  <xs:complexType>
```

```
    <xs:sequence>
      <xs:choice>
        <xs:sequence>
          <xs:element name="configParam" type="variable"/>
          <xs:element name="analysisResult" type="variable" minOccurs="0"/>
        </xs:sequence>
        <xs:sequence>
          <xs:choice>
            <xs:element name="coordinateList" type="mlist"/>
            <xs:element name="pointList" type="plist"/>
          </xs:choice>
        </xs:sequence>
        <xs:sequence>
          <xs:element name="curveTop" type="nurbs"/>
            <xs:element name="curveBot" type="nurbs"/>
        </xs:sequence>
        <xs:element name="shape" type="xs:string"/>
      </xs:choice>
      <xs:element name="provenance" type="provenanceType"/>
    </xs:sequence>
    <xs:attribute name="uID" type="xs:integer"/>
  </xs:complexType>
</xs:element>
```

The geometry model for the airfoil shown in Figure 13, defined by two NURBS curves (top and bottom) with four control points associated with four weights each, follows.

```
<airfoilType>
  <curveTop ncp="4">
    <controlPoints format="Mathematica">
      <definition> { { 0,0,0 }, { 0,0.020,0 }, { 0.25,0.12,0 }, { 1,0,0 } } </definition>
    </controlPoints>
    <knotVector format="Mathematica">
      <definition> { 0,0,0,0,1,1,1,1 } </definition>
    </knotVector>
    <weightVector format="Mathematica">
      <definition> { 1,1,1,1} </definition>
    </weightVector>
    <order>4</order>
  </curveTop>
  <curveBot ncp="4">
    <controlPoints format="Mathematica">
      <definition> { { 0,0,0 }, { 0,-0.005,0 }, { 0.25,-0.04,0 }, { 1,0,0 } } </definition>
    </controlPoints>
    <knotVector format="Mathematica">
      <definition> { 0,0,0,0,1,1,1,1 } </definition>
    </knotVector>
    <weightVector format="Mathematica">
      <definition> { 1,1,1,1 } </definition>>
    </weightVector>
    <order>4</order>
  </curveBot>
</airfoilBase>
```

Table 3

B58 Element Statistics

Element Name	Count
ADML, aircraft, model, airfoils, airfoil, shape, wings, fuselages, fuselage, length, width, hTop, hBot, X0, kLoc, kMag, kWidth, propulsion, engine, weight, global, machNumber, altitude, desiredMeshSize, cowl, N1, N2, topWidth, botWidth, height, length, x0, y0, Nx, Ny, LEAmplifier, thicknessAmplifier, TEAmplifier, ramp, length, width, topHeight, botHeight, x0, y0, kLoc, kMag, materials	1
wing, planform, structure, sections, numRibs, numSpars, semiB, cRoot, cTip, controlEffectors, controlEffector, IBEta, OBEta, chord, Nx, Ny, N1, N2, TEBreak, LEBreak, lambdaLE, lambdaTE, Bu, Bl, shear1, shear2, twist0, twist1, twist2, theta, beta, flapin, flapout, flapchord, inflapdef, outflapdef, orthoTropicMaterial, E1, E2, NU12, G12, RHO, Xt, Xc, Yt, Yc, S, LMType, LThicklb, LThickub	2
provenance	3
t	4
section, crossSection, sparElementIDs, ribElementID, trueRibIDs, ghostRibIDs, airfoilID	18
name	22
v	34
sval	77
description	108

3.4.2. Encoding Convair B58

The Convair B58 Hustler is used as a proof of concept for encoding an entire aircraft in ADML. The reason for using the B58 as the testbed is that most of the data for the B58 is public domain. Also the ADML schema development is in direct response to CCMS needs and the B58 aircraft has been used as a benchmark for design projects in CCMS. The Convair B58 has a delta wing and a vertical tail with four General Electric J79 engines in pods under the wing. The ADML encoding for the B58 is about 700 lines (about 22KB), and uses 111 elements from the total number of 412 ADML elements. Table 3 lists the number of occurrences for those 111 elements that are used for encoding the B58. Owing to the complexity of the ADML schema and the large number of ADML elements required to represent the entire B58 aircraft, it is not feasible to list and discuss each and every element in this thesis. Instead, a detailed description of just one element, the *wing* element for the B58, follows (the full ADML encoding of the B58 is in the supporting files for this thesis).

The wing geometry for the B58 is in VT-CST format. Each *wing* element consists of four subelements, *planform*, *structure*, *controlEffectors*, and *compositeWingBoxes*. The planform is defined by the root chord (*cRoot*), the tip chord (*cTip*), the half span of the wing (*semiB*), standard vectors of leading edge and trailing edge sweep angles (*LESweep* and *TESweep*), and vectors of the nondimensional leading edge and trailing edge break locations (*LEBreak* and *TEBreak*). In addition to these parameters, a *planform* also consists of definitions for the lower and upper surface amplifiers (discussed in great detail in [35]) and a set of corresponding design variables (e.g., N_x and N_y defining the order of the Bernstein polynomials in x and y, respectively, etc.). By storing the leading and trailing edge sweep angles and break locations in a vector, planforms ranging from very simple delta wings to complex wings with multiple breaks and sweeps can be defined. All of the vectors are defined using the *vtype* element from the data schema, and all of the tensors are defined using the *tensor* element from the data schema.

The airfoil cross section for the B58 aircraft, NACA 0003.46-64.069 for the root and NACA 0004.08-63 for the tip chord, is defined as a reference to the *airfoilType* defined outside the *wing* element. The wing *structure* is defined as a set of wing *sections*. Each *section* consists of a unique identifier *uID*, a *name*, a *description*, a reference to a *crossSections*, and definitions for materials and thicknesses for ribs, spars, and skins of a wing.

The *controlEffectors* are defined using VT-CST geometry for control surfaces, a set of parameters for describing a hinge, and the *provenance* element. The *compositeWingBoxes* element defines one or more *compositeWingBox*es, each defined using a set of parameters describing the orientations and the core and layer thicknesses for the ribs, spars, and skins.

A snippet (a subset of the parameters in *planform*, one of the nine *sections* of the main wing of the B58, and the *controlEffectors*) of the ADML wing encoding for the B58 aircraft follows, omitting the schemata defining the tags.

```
<wings>
  <wing uID="1" type1="main" type2="horizontal">
    <planform>
      <semiB>
        <description>(semi span, double)</description>
        <sval>28.4</sval>
      </semiB>
      <cRoot>
        <description>(root chord, double)</description>
        <sval>54.3</sval>
      </cRoot>
      <cTip>
        <description>(tip chord, double)</description>
        <sval>0.01</sval>
      </cTip>
      ...
      <TEBreak>
        <description>(span TE break locations, vector double)</description>
        <sval>n</sval>
      </TEBreak>
      ...
      <Bu>
        <description>(upper surface amp, vector of vector of double)</description>
        <t>
          <v> 1.0 0.0599672208644 0.0504355463712 0.0577872061211 0.0454111489566 1.0 1.0 </v>
          <v> 1.0 0.0680553345373 0.0462947699499 0.0353887592720 0.0562369691330 1.0 1.0 </v>
         <v> 1.0 0.0700420166919 0.00664473683005 0.0507840408258 0.0554809619909 1.0 1.0 </v>
          <v> 1.0 0.0745163222007 0.0245454256447 0.0628921947699 0.0792447631647 1.0 1.0 </v>
           <v> 1.0 0.075914436711 0.0193230458869 0.0591386907617 0.0455063767249 1.0 1.0 </v>
            <v> 1.0 1.0 1.0 1.0 1.0 1.0 1.0 </v>
        </t>
      </Bu>
      ...
    </planform>
    <structure>
      <numRibs>9</numRibs>
      <numSpars>9</numSpars>
      <sections>
        <section uID="1">
          <name>sec1_main_wing</name>
          <description>section 1 of main wing</description>
          <crossSection uID="1">
            <airfoilID>1</airfoilID>
          </crossSection>
          <sparElementIDs>1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
          1 1 1 1 1 1 1 1 1 1 1 1 1</sparElementIDs>
          <ribElementID>2</ribElementID>
          <trueRibIDs>1 4</trueRibIDs>
          <ghostRibIDs>2 3</trueRibIDs>
        </section>
        ...
      </sections>
    </structure>
    <controlEffectors>
      <controlEffector>
        <IBEta>0.1656</IBEta>
        <OBEta>0.6933</OBEta>
        <chord>6.971</chord>
      </controlEffector>
    </controlEffectors>
  </wing>
<wings>
```

3.4.3. Comparison with CPACS

A careful review of CPACS XML schema suggests that the CPACS schema development follows a top-down approach with the most detail at a high level of aircraft design constructs

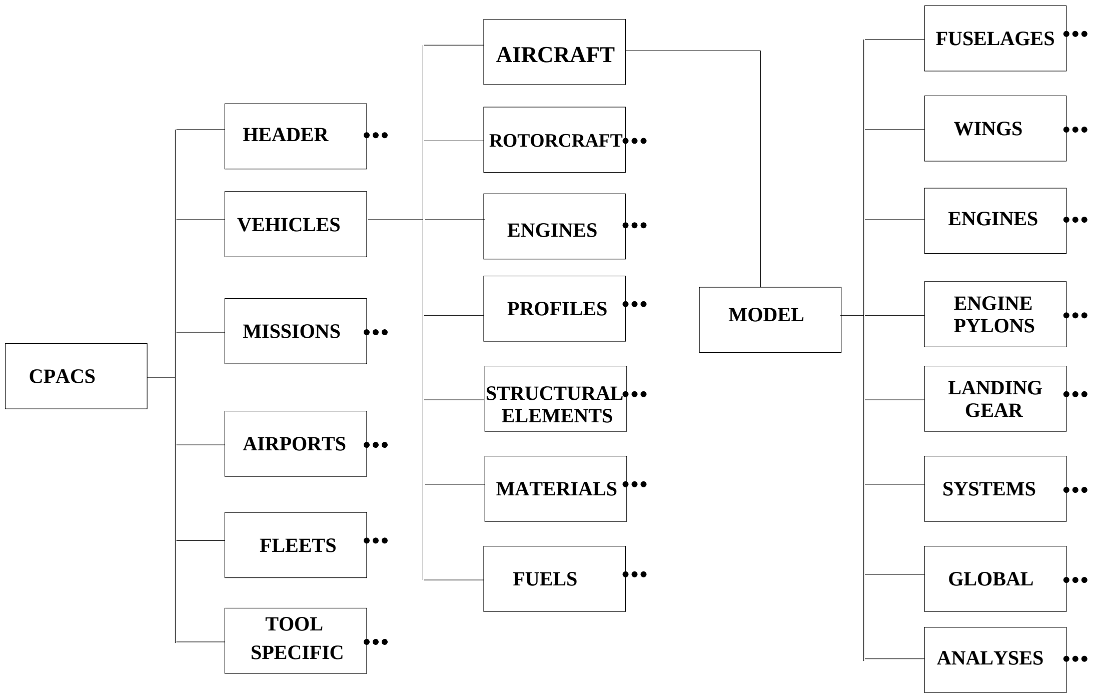


Figure 14. CPACS Taxonomy.

(see Figure 14). On the other hand, ADML follows a bottom-up approach whereby emphasis is given to low level components (raw data, mathematical functions, and basic geometry) that makes it very efficient and expressive. CPACS does not yet support certain detailed geometry, e.g., parametric or NURBS based airfoil design, and the CPACS development team is planning to address those in the next version. In order to support the CST ([33]) based parametric geometry in CPACS, the generated geometry associated with the set of design variables is converted to CPACS format by a special initializer routine. ADML, on the other hand, can *directly* represent the CST input (in VT-CST format) in XML format.

ADML has a very sophisticated way of encoding these constructs, using its low level elements (e.g., ADML has an element called "nurbs", and can directly encode arbitrary parametric functions), and the current version of ADML enables a complete representation of an aircraft. The idea behind having detailed low level schemata for ADML is that once a strong foundation is in place with all common, generic, reusable elements at a low level, one can easily build upon those all other high level aircraft design components with the flexibility to accommodate several different configurations. Careful thought has been given to a format for representing mathematical objects while developing the proposed ADML schema. A major difference between CPACS and ADML, as mentioned earlier, is the representation of low level data elements. CPACS does not have any provision in the current version for representing a mathematical function in analytical form. All the *profile* elements (e.g., fuselage and wing cross sections) in CPACS are defined as *pointList*s. Earlier versions of CPACS defined each *pointList* as a sequence of three XML elements defining three coordinate axes. This verbose definition of a list data type incurs significant overhead in terms of the storage of the XML documents. The *pointList* definition has been modified in the current version of CPACS so that a list of points along a coordinate axis is represented as a vector. Another peculiarity of CPACS is that both the vector and the array data types are defined as XML strings (an array is a flattened list), and there is no data structure in CPACS to handle multidimensional arrays. Reshaping a matrix from a string significantly increases the cost for parsing the CPACS data.

CPACS has been adopted as a data standard for exchanging aircraft design and analysis data in several DLR projects and integrated environments, and a number of tools (e.g., TXL—a geometry engine) have been developed for automating the multidisciplinary aircraft design process. ADML is still in an early stage of development, and has to evolve further to accommodate a wide variety of aircraft configurations (CPACS has about two thousand five hundred elements whereas ADML has about four hundred elements) The immediate goal is the adoption of ADML within the CCMS, and ultimately within a large segment of the aircraft design community. Another possibility is to merge CPACS and ADML to achieve the benefits of both. This could be achieved by using the existing import facility available in the XML Schema definition. The XML Schema *import* element facilitates adding multiple schemata with different target namespaces to an existing schema. That way CPACS schema could be imported into the ADML schema, and all the elements in CPACS could be accessed through ADML without any difficulty.

3.5. Conclusion and Future Work

An XML schema based generic, comprehensive, and compact aircraft design markup language (ADML) is proposed to represent aircraft design models (geometry, structures, propulsion, etc.) and analysis data (raw data and mathematical objects). ADML addresses data exchange and interoperability issues involved in a multidisciplinary, collaborative, conceptual aircraft design environment by providing a common language for data communication. The XML schema discussed in this thesis follows a modular schema development and takes a bottom up approach by starting the schema development from the simplest form of data and building on that more complex constructs in a conceptual aircraft design process. Thus, the XML elements from the data and function schema serve as the building blocks for other more complex elements. An airfoil geometry example presented in Section 4 illustrates the modeling capabilities of the proposed geometry schema, and the aircraft model represented using the Convair B58 shows the scope of the proposed schema. ADML supports both the VT-CST geometry as well as NURBS based BRep constructs. The ADML schema supports several disciplines (geometry, structures, configuration layout, propulsion, mission, performance, payload, and materials) involved in a multidisciplinary, collaborative conceptual aircraft design and analysis process where all disciplines can natively understand the ADML standard and can communicate with each other through a common language and platform neutral data format. The schema described in this thesis is organized for the design and analysis of fixed wing aircraft, but it is readily extensible to flapping wing MAVs (micro air vehicles) and morphing vehicles, whose shapes change in time. ADML is still in an early stage of development, and has to evolve further to accommodate a wide variety of aircraft configurations, though ADML is complete enough to represent an entire B58 used as a conceptual design benchmark by CCMS and others.

Chapter 4.

CONCLUSION AND FUTURE WORK

Two computational tools are proposed in this thesis to solve specific problems encountered in today's modern conceptual aircraft design and analysis, and to advance the state-of-the-art in the multidisciplinary conceptual aircraft design process. Chapter 1 introduced the topic and motivated the need for these computational tools. Chapter 2 provided a detailed description of a new multiobjective optimization algorithm. In particular, it discussed how the proposed method fits into the multidisciplinary conceptual aircraft design optimization process, established the mathematical foundation for the proposed methodology, and described the general scheme of the optimization algorithm. Chapter 3 provided a detailed description of the XML schema based markup language ADML.

The proposed Pareto front approximation method from the first part of the thesis generates a good collection of well-distributed Pareto optimal solutions for diverse types of problems. Another contribution of this part is the generalization of the star discrepancy based performance measure (introduced as a new quantitative measure of point set distribution) for problems with more than two objectives. The method compares very favorably to other deterministic multiobjective optimization methods in those quantitative measures. The ultimate goal is to apply the proposed method to complex multidisciplinary design optimization problems where solving blackbox simulation based multiobjective optimization problems with possibly nonsmooth functions is prevalent. Another extension of the method would be to consider problems with constraints. Yet another application of the proposed method would be for solving machine learning problems, constrained clustering in particular.

The second part of the thesis deals with the interoperability issues involved in a collaborative, multidisciplinary conceptual aircraft design and analysis environment. An XML schema based markup language ADML is proposed as a common language for data communication. An important feature of ADML is the very expressive low level representation of all the common components such as raw data, mathematical objects, and basic geometry. As a proof of concept the ADML schema is used to encode an entire Convair B58. The future plan is to extend the ADML schema to support several different nonconventional (flying wing, joined wing, flapping wing, truss-braced wing, dirigible) aircraft configurations.

REFERENCES

[1] Aerospace and Defence Industries Associations Europe, "Standard for the Exchange of Product model data (STEP - ISO 10303) Application Protocol 209: Multidisciplinary analysis and design", 2013.

[2] Amenta N., Attali D., and Devillers O., "Complexity of Delaunay triangulation for points on lower-dimensional polyhedra", in *Proceedings of 18th annual ACM-SIAM SYMPOS. on Discrete Algorithms*, pp. 1106–1113, 2007.

[3] American Institute of Aeronautics and Astronautics: Flight dynamics model exchange standard (draft), "BSR/AIAA S-119-201X", AIAA, 2010.

[4] Arora J., *Introduction to optimum design*, Academic Press, pp. 429, 2004.

[5] Audet C. and Dennis J. E., "Mesh adaptive direct search algorithms for constrained optimization", *SIAM Journal on Optimization*, vol. 17, No. 1, pp. 188–217, 2006.

[6] Audet C., Savard G., and Zghal W., "Multiobjective optimization through a series of single objective formulations", *SIAM Journal of Optimization*, vol. 19, pp. 188–210, 2008.

[7] Audet C., Savard G., and Zghal W., "A mesh adaptive direct search algorithm for multiobjective optimization", *European Journal of Operational Research*, vol. 204, pp. 545–556, 2010.

[8] Begley E. F. and Sturrock C. P., "MatML: XML for Material Property Data", in *ASM Internationals Advanced Materials and Processes*, available online at http:// xml.coverpages.org / begley-ampmatml.pdf, 2000.

[9] Berndt J. S., "JSBSim, an open source platform independent flight dynamics model in C++", JSBSim Reference Manual v1.0., available online at http://jsbsim.sourceforge.net/ JSBSimReferenceManual.pdf, 2011.

[10] Bhandarkar M. P. and Nagi R., "STEP-based feature extraction from STEP geometry for Agile Manufacturing", *Computers in Industry*, vol. 41, pp. 3-24, 2000.

[11] Blair M., "Air Vehicle Environment in C++: A Computational Design Environment for Conceptual Innovations", in *Journal of Aerospace Computing, Information, and Communication*, vol. 7, pp. 85-117, 2010.

[12] Caflisch R. E., "Monte Carlo and quasi-Monte Carlo methods", *Acta Numerica*, vol. 7, pp. 1–49, 1998.

[13] Composite Materials and Structures Group, "FemML for Data Exchange between FEA Codes", in *ANSYS Users group conference, Univ. of Maryland, College Park*, available online at http://femml.sourceforge.net/, Oct. 2001.

[14] Das I. and Dennis J.E., "Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems ", *SIAM Journal on Optimization*, vol. 8. pp. 631–657, 1998.

[15] Deb K., "Multiobjective genetic algorithms: problem difficulties and construction of test problems", *Evol. Comput.*, vol. 7. pp. 205–230, 1999.

[16] Deb K., Pratap. A., Agarwal S., and Meyarivan T., "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Transaction on Evolutionary Computation*, vol. 6(2), pp. 181–197, 2002.

[17] Deb K., Thiele L., Laumanns., and Zitzler E., "Scalable Test Problems for Evolutionary Multiobjective Optimization", TIK–Technical Report No. 112 Institut fur Technische Informatik und Kommunikationsnetze, ETH Zurich, Switzerland, 2001.

[18] Deshpande S. G., Watson L. T., and Canfield R. A., "Biobjective Optimization using Direct Search Techniques ", in *IEEE SoutheastCon*, Jacksonville, FL, Apr. 2013.

[19] Deshpande S. G., Watson L. T., and Canfield R. A., "Pareto Front Approximation using a Hybrid Approach ", in *Procedia Computer Science*, vol. 18, pp. 521–530, 2013.

[20] Deshpande S. G., Watson L. T., Canfield R.. A., Blair M., and Beran P. S. , "XML Schema for Aircraft Conceptual Model Representation", in *International Conference on Information and Knowledge Engineering*, Las Vegas, Nevada, 2011.

[21] Digabel S. L. and Tribes C., *NOMAD user guide version 3.5.1, Les cahiers du GERAD, G200937*, Available at http://www.gerad.ca/NOMAD/Downloads/user_guide.pdf, 2009.

[22] Eichfelder G., *Adaptive Scalarization Methods in Multiobjective Optimization (Vector Optimization)*, Springer, 2008.

[23] Gopalsamy S. and Yu T., "A Geometry Engine for CAD/GRID Integration", in *AIAA 2003-800, 41st Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, Jan. 2003.

[24] Gray G. A. and Fowler K. R., "Traditional and hybrid derivative-free optimization approaches for black box functions", *Computational Optimization, Methods and Algorithms*, Vol. 356, pp. 125–151, 2011.

[25] Haimes R. and Dannenhoffer J. F., "Control of Boundary Representation Topology in Multidisciplinary Analysis and Design", in *AIAA 2010-1504, 48th AIAA Aerospace Sciences Meeting*, Orlando, Florida, Jan. 2010.

[26] He J., Watson L. T., Ramakrishnan N., Shaffer C. A., Verstak A., Jiang J., Bae K., and Tranter W. H., "Dynamic data structures for a direct search algorithm", *Computational Optimization and Applications*, Vol. 23, pp. 5–25, 2002.

[27] Hucka M. et al., "The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models", *Bioinformatics*, Vol. 19 (4), pp. 524–531, 2003.

[28] Information Processing – Text and Office Systems, *Standard Generalized Markup Language (SGML)*, ISO 8879:1986.

[29] Initial Graphics Exchange Specifications, "A Century of Excellence in Measurements, Standards, and Technology—A Chronicle of Selected NBS/NIST Publications, 1901 - 2000, David L. Lide, Editor", NIST Special Publication 958, Jan. 2001.

[30] Jones D. R., Perttunen C. D., and Stuckman B. E., "Lipschitzian optimization without the Lipschitz constant", *Journal of Optimization Theory and Application*, Vol. 79, No. 1, pp. 157–181, 1993.

[31] Krishnan R., "Evaluation of Frameworks for HSCT Design and Optimization ", NASA/CR-1998-208731, Oct. 1998.

[32] Kugele S. C., Watson L. T., and Trosset M. W., "Multidimensional numerical integration for robust design optimization", in *ACM Southeast Regional Conference*, pp. 385–390, 2007.

[33] Kulfan, B. M., "Universal Parametric Geometry Representation Method", in *Journal of Aircraft, Vol. 45, No. 1, pp. 142158*, Jan. 2008.

[34] Lin R. and Afjeh A., "An extensible, interchangeable, and sharable database model for improving multidisciplinary aircraft design", in *AIAA 2002-5613*, Atlanta, GA, 2002.

[35] Morris C. C., Allison D. L., Schetz J. A., and Kapania R. K., "Parametric geometry model for multidisciplinary design optimization of tailless supersonic aircraft", in *Proceedings of AIAA Modeling and Simulation Technologies Conference*, Minneapolis, Minnesota, Aug. 2012.

[36] Nagel R. N., Braithwaite W. W., and Kennicott P. R., "Initial Graphics Exchange Specification IGES, Version 1.0", Washington DC: National Bureau of Standards, NBSIR 80-1978, 1980.

[37] Peak R., Lubell J., Srinivasan V., and Waterbury S. C., "STEP, XML, and UML: Complementary Technologies", in *ASME, Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Salt Lake City, USA, 2004.

[38] Pratt M., "Extension of ISO 10303: The Step Standard, for the Exchange of Procedural Shape Models", in *Proc. Int'l Conf Shape Modeling and Applications (SMI)*, 2004.

[39] Rappoport A., "An Architecture for Universal CAD Data Exchange", in *Proceedings, Solid Modeling 03*, SCM Press, Seattle, WA, Jun. 2003.

[40] Raymer D. P., *Aircraft Design: A Conceptual Approach*, AIAA Education Series, New York, NY, 2006.

[41] Rizzi A., Oppelstrup J., Zhang M., and Tomac M., "Coupling Parametric Aircraft Lofting to CFD and CSM Grid Generation for Conceptual Design", in *AIAA 2011-160, 49th AIAA Aerospace Sciences Meeting*, Orlando, Florida, Jan. 2011.

[42] Roth G. L., Livingston J. W., Blair M., and Kolonay R., "CREATE-AV DaVinci: Computationally based engineering for conceptual design", in *AIAA 2010-1232*, Orlando, FL, Jan. 2010.

[43] Ryu J., Kim S., and Wan H., "Pareto front approximation with adaptive weighted sum method in multiobjective simulation optimization", in *Proceedings of the 2009 Winter Simulation Conference*, pp. 623–633, Austin, TX, USA, 2009.

[44] Salas, A. O., and Townsend, J. C., "Framework Requirements for MDO Application Development", in *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA Paper 98-4740, Sep. 1998.

[45] Sandhu R., *The MathML Handbook*, Charles River Media, Edition 1, 2002.

[46] Schenck D. and Wilson P., *Information Modeling the EXPRESS Way*, Oxford University Press, 1994.

[47] Thacker W. I., Zhang J, Watson L. T., Birch J. B., Iyer M. A., Barry M. W., "Algorithm 905: SHEPPACK: modified Shepard algorithm for interpolation of scattered multivariate data", *ACM Trans. Math. Software*, vol. 37, pp. 1–20, 2010.

[48] Veldhuizen D. A. and Lamont G. B, *Evolutionary Computation and Convergence to a Pareto Front*, Stanford University, California, Morgan Kaufmann publication, pp. 221–228, 1998.

[49] Wang L., Ishida H., Hiroyashu T., and Miki M., "Examination of multiobjective optimization method for global search using DIRECT and GA", in *IEEE World Congress on Computational Intelligence*, pp 2446–2451, 2008.

[50] World Wide Web Consortium, *XML Schema Part 0: Primer*, Available online at http://www.w3.org/TR/xmlschema-0, May 2001.

[51] World Wide Web Consortium, *Extensible Markup Language (XML) 1.0, 3rd ed.*, , available online at http://www.w3.org/TR/2004/REC-xml-20040204/, Feb. 2004.

Appendix A: ADML Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="ADML">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="data"/>
      <xs:element ref="functions"/>
      <xs:element ref="geometry"/>
      <xs:element ref="aircraft"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="data">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="variable"/>
      <xs:element ref="flag"/>
      <xs:element ref="scalar"/>
      <xs:element ref="tensor"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="nd">
  <xs:sequence>
    <xs:element name="name" type="xs:string" minOccurs="0"/>
    <xs:element name="description" type="xs:string" minOccurs="0"/>
    <xs:element name="unit" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="scalarType">
  <xs:union memberTypes="xs:integer xs:double xs:string xs:boolean"/>
</xs:simpleType>
<xs:element name="min" type="scalarType"/>
<xs:element name="max" type="scalarType"/>
<xs:simpleType name="flagType">
  <xs:union memberTypes="xs:integer xs:boolean"/>
</xs:simpleType>
<xs:element name="flag">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="nd">
        <xs:sequence>
          <xs:element name="flagval" type="flagType"/>
        </xs:sequence>
      </xs:extension>
     </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:simpleType name="vtype">
   <xs:list itemType="scalarType"/>
</xs:simpleType>
<xs:complexType name="tensorType">
  <xs:choice>
    <xs:sequence minOccurs="2" maxOccurs="unbounded">
      <xs:element name="t" type="tensorType"/>
    </xs:sequence>
    <xs:sequence minOccurs="2" maxOccurs="unbounded">
      <xs:element name="v" type="vtype"/>
    </xs:sequence>
  </xs:choice>
</xs:complexType>
<xs:element name="value" type="valueType"/>
<xs:complexType name="valueType">
  <xs:choice>
    <xs:element ref="scalar"/>
    <xs:element ref="tensor"/>
  </xs:choice>
</xs:complexType>
<xs:element name="scalar">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="nd">
        <xs:sequence>
          <xs:element name="sval" type="scalarType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element name="tensor">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="nd">
        <xs:sequence>
          <xs:element name="t" type="tensorType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

```
<xs:element name="variable">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="nd">
        <xs:sequence>
          <xs:element ref="value"/>
          <xs:element ref="min" minOccurs="0"/>
          <xs:element ref="max" minOccurs="0"/>
          <xs:element ref="flag" minOccurs="0"/>
          <xs:element ref="scalar" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="vid" type="xs:ID"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element name="functions">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="operator"/>
      <xs:element ref="expression"/>
      <xs:element ref="relation"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="arguments" type="xs:string"/>
<xs:element name="variables" type="xs:string"/>
<xs:element name="listRef" type="xs:IDREF"/>
<xs:element name="expressionRef" type="xs:IDREF"/>
<xs:element name="definition" type="xs:string"/>
<xs:simpleType name="formatType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Matlab"/>
    <xs:enumeration value="Mathematica"/>
  </xs:restriction>
</xs:simpleType>
<xs:attribute name="format" type="formatType"/>
<xs:attribute name="structure" type="structureType"/>
<xs:simpleType name="structuretype">
  <xs:restriction base="xs:string">
    <xs:enumeration value="array"/>
    <xs:enumeration value="general"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="operator">
  <xs:complexType>
```

```
    <xs:complexContent>
      <xs:extension base="nd">
        <xs:sequence>
          <xs:element name="domain" type="xs:string"/>
          <xs:element name="range" type="xs:string"/>
          <xs:element ref="arguments"/>
          <xs:element ref="definition"/>
        </xs:sequence>
        <xs:attribute ref="format"/>
        <xs:attribute name="opid" type="xs:ID"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element name="expression">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="nd">
        <xs:sequence>
          <xs:element ref="variables"/>
          <xs:element ref="definition"/>
          <xs:element name="patternStr" type="xs:string" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute ref="format"/>
        <xs:attribute name="eid" type="xs:ID"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:complexType name="mlist">
  <xs:complexContent>
    <xs:extension base="nd">
      <xs:sequence>
        <xs:element ref="variables"/>
        <xs:element ref="definition"/>
      </xs:sequence>
      <xs:attribute name="lid" type="xs:ID"/>
      <xs:attribute ref="format"/>
      <xs:attribute ref="structure"/>
      <xs:attribute name="dimension" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="relation">
  <xs:complexType>
    <xs:complexContent>
```

```
      <xs:extension base="nd">
        <xs:sequence>
          <xs:choice>
            <xs:element ref="expressionRef"/>
            <xs:element ref="listRef"/>
          </xs:choice>
        </xs:sequence>
        <xs:attribute ref="format"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:complexType name="geometry">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element name="point" type="point"/>
        <xs:element name="plist" type="pList"/>
        <xs:element name="line" type="line"/>
        <xs:element name="plane" type="plane"/>
        <xs:element name="nurbs" type="nurbs"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="point">
  <xs:list itemType="xs:double"/>
</xs:simpleType>
<xs:element name="pList">
  <xs:complexType>
    <xs:sequence maxOccurs="unbounded">
      <xs:element name="p" type="point"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="line">
  <xs:sequence>
    <xs:element name="point1" type="point"/>
    <xs:element name="point2" type="point"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="plane">
  <xs:sequence>
    <xs:element name="P0" type="point"/>
    <xs:element name="normal" type="point"/>
  </xs:sequence>
```

```
</xs:complexType>
<xs:element name="frame">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="origin" type="point"/>
      <xs:element name="angles" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="frameID" type="xs:ID"/>
    <xs:attribute name="parentID" type="xs:IDREF"/>
  </xs:complexType>
</xs:element>
<xs:simpleType name="nurbsType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="curve"/>
    <xs:enumeration value="surface"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="nurbs">
  <xs:complexContent>
    <xs:extension base="nd">
      <xs:sequence>
        <xs:element name="controlPoints" type="mlist"/>
        <xs:element name="knotVector" type="mlist"/>
        <xs:element name="weightVector" type="mlist"/>
        <xs:element name="order" type="xs:integer"/>
      </xs:sequence>
      <xs:attribute name="nurbsID" type="xs:ID"/>
      <xs:attribute name="ntype" type="nurbsType" minOccurs="0"/>
      <xs:attribute name="ncp" type="xs:integer" minOccurs="0"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="provenanceType" type="xs:string"/>
<xs:element name="aircraftGlobalType">
  <xs:complexType>
    <xs:complexContent>
      <xs:sequence>
        <xs:element name="machNumber" type="scalar"/>
        <xs:element name="altitude" type="scalar"/>
        <xs:element name="desiredMeshSize" type="scalar"/>
      </xs:sequence>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element name="airfoilType">
  <xs:complexType>
```

```
    <xs:sequence>
      <xs:choice>
        <xs:sequence>
          <xs:element name="configParam" type="variable"/>
          <xs:element name="analysisResult" type="variable"
          minOccurs="0"/>
        </xs:sequence>
        <xs:sequence>
          <xs:choice>
            <xs:element name="coordinateList" type="mlist"/>
            <xs:element name="pointList" type="plist"/>
          </xs:choice>
        </xs:sequence>
        <xs:sequence>
          <xs:element name="curveTop" type="nurbs"/>
          <xs:element name="curveBot" type="nurbs"/>
        </xs:sequence>
        <xs:element name="shape" type="xs:string"/>
      </xs:choice>
      <xs:element name="provenance" type="provenanceType"/>
    </xs:sequence>
    <xs:attribute name="uID" type="xs:integer" use="required"/>
  </xs:complexType>
</xs:element>
<xs:complexType name="SEIType">
  <xs:complexContent>
    <xs:extension base="nd">
      <xs:sequence>
        <xs:element name="ownership" type="xs:string" minOccurs="0"/>
        <xs:element name="relationship" type="xs:string" minOccurs="0"/>
        <xs:element name="material" type="xs:integer" minOccurs="0"/>
        <xs:element name="thicknesses" type="vtype" minOccurs="0"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:sequence>
      <xs:attribute name="ID" type="xs:integer" use="required"/>
    </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="SECType">
  <xs:complexContent>
    <xs:extension base="nd">
      <xs:sequence>
        <xs:element name="ownership" type="xs:string" minOccurs="0"/>
        <xs:element name="relationship" type="xs:string" minOccurs="0"/>
        <xs:element name="laminaAngles" type="vtype" minOccurs="0"/>
        <xs:element name="laminaThicknesses" type="vtype" minOccurs="0"/>
        <xs:element name="coreThickness" type="xs:double" minOccurs="0"/>
        <xs:element name="laminaMaterials" type="vtype" minOccurs="0"/>
        <xs:element name="coreMaterial" type="xs:integer" minOccurs="0"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:sequence>
      <xs:attribute name="ID" type="xs:integer" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="structuralElementsType">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="structElementIsotropic" type="SEIType"
      maxOccurs="unbounded"/>
      <xs:element name="structElementComposite" type="SECType"
      maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="isotropicMaterialType">
  <xs:complexContent>
    <xs:extension base="nd">>
      <xs:sequence>
        <xs:element name="type" type="xs:string" minOccurs="0"/>
        <xs:element name="E" type="xs:double" minOccurs="0"/>
        <xs:element name="nu" type="xs:double" minOccurs="0"/>
        <xs:element name="rho" type="xs:double" minOccurs="0"/>
        <xs:element name="yield" type="xs:double" minOccurs="0"/>
        <xs:element name="ThicknessLB" type="xs:double" minOccurs="0"/>
        <xs:element name="ThicknessUB" type="xs:double" minOccurs="0"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:sequence>
      <xs:attribute name="ID" type="xs:integer" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType  name="compositeMaterialType">
  <xs:complexContent>
    <xs:extension base="nd">>
      <xs:sequence>
        <xs:element name="design" type="xs:string" minOccurs="0"/>
        <xs:element name="symmetric" type="xs:string" minOccurs="0"/>
        <xs:element name="core" type="xs:integer" minOccurs="0"/>
        <xs:element name="coreMat" type="xs:string" minOccurs="0"/>
        <xs:element name="layerMat" type="xs:string" minOccurs="0"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:sequence>
      <xs:attribute name="ID" type="xs:integer" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="orthotropicMaterialType">
  <xs:complexContent>
    <xs:extension base="nd">>
      <xs:sequence>
        <xs:element name="E1" type="xs:double" minOccurs="0"/>
        <xs:element name="E2" type="xs:double" minOccurs="0"/>
        <xs:element name="NU12" type="xs:double" minOccurs="0"/>
        <xs:element name="G12" type="xs:double" minOccurs="0"/>
        <xs:element name="G1z" type="xs:double" minOccurs="0"/>
        <xs:element name="G2Z" type="xs:double" minOccurs="0"/>
        <xs:element name="RHO" type="xs:double" minOccurs="0"/>
        <xs:element name="Xt" type="xs:double" minOccurs="0"/>
        <xs:element name="Xc" type="xs:double" minOccurs="0"/>
        <xs:element name="Yt" type="xs:double" minOccurs="0"/>
        <xs:element name="Yc" type="xs:double" minOccurs="0"/>
        <xs:element name="S" type="xs:double" minOccurs="0"/>
        <xs:element name="F12" type="xs:double" minOccurs="0"/>
        <xs:element name="STRN" type="xs:double" minOccurs="0"/>
        <xs:element name="LMType" type="xs:double" minOccurs="0"/>
        <xs:element name="LThicklb" type="xs:double" minOccurs="0"/>
        <xs:element name="LThickub" type="xs:double" minOccurs="0"/>
        <xs:element name="shearAllowBond" type="xs:double" minOccurs="0"/>
        <xs:element name="NSM" type="xs:double" minOccurs="0"/>
        <xs:element name="type" type="xs:string" minOccurs="0"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:sequence>
      <xs:attribute name="ID" type="xs:integer" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="materialsType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">>
      <xs:sequence>
        <xs:element name="isotropicMaterial" type="isotropicMaterialType"
        minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="compositeLaminate" type="compositeMaterialType"
        minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="orthotropicMaterial" type="orthotropicMaterialType"
        minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="fuselageType">
  <xs:complexContent>
    <xs:extension base="nd">>
      <xs:sequence>
        <xs:element name="NC" type="scalar" minOccurs="0"/>
        <xs:element name="ND" type="scalar" minOccurs="0"/>
        <xs:element name="length" type="scalar" minOccurs="0"/>
        <xs:element name="width" type="scalar" minOccurs="0"/>
        <xs:element name="hTop" type="scalar" minOccurs="0"/>
        <xs:element name="hBot" type="scalar" minOccurs="0"/>
        <xs:element name="x0" type="scalar" minOccurs="0"/>
        <xs:element name="kLoc" type="scalar" minOccurs="0"/>
        <xs:element name="kMag" type="scalar" minOccurs="0"/>
        <xs:element name="kWidth" type="scalar" minOccurs="0"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:sequence>
      <xs:attribute name="uID" type="xs:integer" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="fuselagesType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="fuselage"
        type="fuselageType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="distributedMassType">
  <xs:complexContent>
    <xs:extension base="nd">
      <xs:sequence>
        <xs:element name="density" type="xs:double"/>
        <xs:element name="gain" type="xs:double"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="planformType">
  <xs:complexContent>
    <xs:extension base="nd">
```

```
      <xs:sequence>
        <xs:element name="location" type="point" minOccurs="0"/>
        <xs:element name="semiB" type="scalar" minOccurs="0"/>
        <xs:element name="cRoot" type="scalar" minOccurs="0"/>
        <xs:element name="cTip" type="scalar" minOccurs="0"/>
        <xs:element name="Nx" type="scalar" minOccurs="0"/>
        <xs:element name="Ny" type="scalar" minOccurs="0"/>
        <xs:element name="N1" type="scalar" minOccurs="0"/>
        <xs:element name="N2" type="scalar" minOccurs="0"/>
        <xs:element name="LEBreak" type="vtype" minOccurs="0"/>
        <xs:element name="TEBreak" type="vtype" minOccurs="0"/>
        <xs:element name="LESweep" type="vtype" minOccurs="0"/>
        <xs:element name="TESweep" type="vtype" minOccurs="0"/>
        <xs:element name="lambdaLE" type="vtype" minOccurs="0"/>
        <xs:element name="lambdaTE" type="vtype" minOccurs="0"/>
        <xs:element name="Bu" type="tensor" minOccurs="0"/>
        <xs:element name="Bl" type="tensor" minOccurs="0"/>
        <xs:element name="shear1" type="scalar" minOccurs="0"/>
        <xs:element name="shear2" type="scalar" minOccurs="0"/>
        <xs:element name="twist0" type="scalar" minOccurs="0"/>
        <xs:element name="twist1" type="scalar" minOccurs="0"/>
        <xs:element name="twist2" type="scalar" minOccurs="0"/>
        <xs:element name="theta" type="scalar" minOccurs="0"/>
        <xs:element name="beta" type="scalar" minOccurs="0"/>
        <xs:element name="flapin" type="scalar" minOccurs="0"/>
        <xs:element name="flapout" type="scalar" minOccurs="0"/>
        <xs:element name="flapchord" type="scalar" minOccurs="0"/>
        <xs:element name="inflapdef" type="scalar" minOccurs="0"/>
        <xs:element name="outflapdef" type="scalar" minOccurs="0"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:sequence>
      <xs:attribute name="uID" type="xs:integer" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="wingCrossSectionType">
  <xs:complexContent>
    <xs:extension base="nd">
      <xs:element name="airfoilID" type="xs:Integer"/>
      <xs:attribute name="uID" type="xs:integer" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="wingSectionType">
  <xs:complexContent>
    <xs:extension base="nd">
```

```
      <xs:sequence>
        <xs:element name="crossSection" type="wingCrossSectionsType"/>
        <xs:element name="sparElementIDs" type="vtype"/>
        <xs:element name="ribElementID" type="xs:integer"/>
        <xs:element name="skinElementIDs" type="vtype"/>
        <xs:element name="trueRibIDs" type="vtype"/>
        <xs:element name="ghostRibIDs" type="vtype"/>
        <xs:element name="sparCapMat" type="vtype"/>
        <xs:element name="ribCapMat" type="vtype"/>
        <xs:element name="ribCapT" type="vtype"/>
        <xs:element name="sparCapT" type="vtype"/>
        <xs:element name="distributedMass" type="distributedMassType"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="wingSectionsType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element name="section" type="wingSectionType"
        minOccurs="2" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="wingStructureType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element name="numRibs" type="xs:integer"/>
        <xs:element name="numSpars" type="xs:integer"/>
        <xs:element name="sections" type="wingSectionsType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="wingType">
  <xs:complexContent>
    <xs:extension base="nd">
      <xs:sequence>
        <xs:element name="planform" type="planformType"/>
        <xs:element name="structure" type="wingStructureType"/>
        <xs:element name="controlEffectors"
        type="controlEffectorsType" minOccurs="0"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:sequence>
      <xs:attribute name="uID" type="xs:integer" use="required"/>
      <xs:attribute name="type1" use="required"/>
      <xs:attribute name="type2" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="engineType">
  <xs:complexContent>
    <xs:extension base="nd">
      <xs:all>
        <xs:element name="cruiseThrust" type="scalar"/>
        <xs:element name="maxBypassRatio" type="scalar"/>
        <xs:element name="fanPressureRatio" type="scalar"/>
        <xs:element name="LPCPR" type="scalar"/>
        <xs:element name="HPCPR" type="scalar"/>
        <xs:element name="HMNBRC" type="scalar"/>
        <xs:element name="LMNBRC" type="scalar"/>
        <xs:element name="loc" type="tensor"/>
        <xs:element name="length" type="scalar"/>
        <xs:element name="width" type="scalar"/>
        <xs:element name="height" type="scalar"/>
        <xs:element name="weight" type="scalar"/>
        <xs:element name="inertiaTensor" type="tensor"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:all>
      <xs:attribute name="uID" type="xs:integer" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="rampType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:all>
        <xs:element name="length" type="scalar"/>
        <xs:element name="width" type="scalar"/>
        <xs:element name="topHeight" type="scalar"/>
        <xs:element name="botHeight" type="scalar"/>
        <xs:element name="x0" type="scalar"/>
        <xs:element name="y0" type="scalar"/>
        <xs:element name="kLoc" type="scalar"/>
        <xs:element name="kMag" type="scalar"/>
        <xs:element name="kWidth" type="scalar"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:all>
      <xs:attribute name="uID" type="xs:integer" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="cowlType">
  <xs:complexContent>
    <xs:extension base="nd">
      <xs:all>
        <xs:element name="N1" type="scalar"/>
        <xs:element name="N2" type="scalar"/>
        <xs:element name="topWidth" type="scalar"/>
        <xs:element name="botWidth" type="scalar"/>
        <xs:element name="height" type="scalar"/>
        <xs:element name="length" type="scalar"/>
        <xs:element name="x0" type="scalar"/>
        <xs:element name="y0" type="scalar"/>
        <xs:element name="Nx" type="scalar"/>
        <xs:element name="Ny" type="scalar"/>
        <xs:element name="LEAmplifier" type="scalar"/>
        <xs:element name="thicknessAmplifier" type="scalar"/>
        <xs:element  name="TEAmplifier" type="scalar"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:all>
      <xs:attribute name="uID" type="xs:integer" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="EEWSType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:all>
        <xs:element name="loc" type="vtype"/>
        <xs:element name="length" type="xs:double"/>
        <xs:element name="width" type="xs:double"/>
        <xs:element name="height" type="xs:double"/>
        <xs:element name="weight" type="xs:double"/>
        <xs:element name="inertiaTensor" type="tensor"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:all>
      <xs:attribute name="uID" type="xs:integer" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="enginesType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
```

```
      <xs:sequence>
        <xs:element name="engine" type="engineType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="cowlsType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element name="cowl" type="cowlType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="rampsType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element name="ramp" type="rampType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="EEWSsType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element name="EEWS" type="EEWSType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="propulsionType">
  <xs:complexContent>
    <xs:extension base="nd">
      <xs:sequence>
        <xs:element name="engines" type="enginesType"/>
        <xs:element name="cowls" type="cowlsType"/>
        <xs:element name="ramps" type="rampsType"/>
        <xs:element  name="EEWSs" type="EEWSsType"/>
      </xs:sequence>
      <xs:attribute name="uID" type="xs:integer" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```
<xs:complexType name="wingsType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element name="wing" type="wingType"  maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="hingeType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element name="location" type="xs:double"/>
        <xs:element name="verticalLocation" type="xs:string"/>
        <xs:element name="type" type="xs:string"/>
        <xs:element name="clamShell" type="xs:integer"/>
        <xs:element name="defMax" type="xs:double"/>
        <xs:element name="stiffness" type="xs:double"/>
        <xs:element name="momentMax" type="xs:double"/>
        <xs:element name="addedMass" type="xs:double"/>
      </xs:sequence>
      <xs:attribute name="uID" type="xs:integer" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="controlEffectorType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element name="IBEta" type="xs:double"/>
        <xs:element name="OBEta" type="xs:double"/>
        <xs:element name="location" type="xs:string"/>
        <xs:element name="chord" type="xs:double"/>
        <xs:element name="hinge" type="hingeType"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:sequence>
      <xs:attribute name="uID" type="xs:integer" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="controlEffectorsType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element name="controlEffector"
        type="controlEffectorType" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="configOptionType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="tricycle"/>
    <xs:enumeration value="quadricycle"/>
    <xs:enumeration value="multiBogey"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="landingGearType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element name="mass" type="vtype"/>
        <xs:element name="coordXLowered" type="vtype"/>
        <xs:element name="coordYLowered" type="vtype"/>
        <xs:element name="coordZLowered" type="vtype"/>
        <xs:element name="coordXRaised" type="vtype"/>
        <xs:element name="coordYRaised" type="vtype"/>
        <xs:element name="coordZRaised" type="vtype"/>
        <xs:element name="coordIxxLowered" type="vtype"/>
        <xs:element name="coordIyyLowered" type="vtype"/>
        <xs:element name="coordIzzLowered" type="vtype"/>
        <xs:element name="coordIxyLowered" type="vtype"/>
        <xs:element name="coordIxzLowered" type="vtype"/>
        <xs:element name="coordIyzLowered" type="vtype"/>
        <xs:element name="coordIxxRaised" type="vtype"/>
        <xs:element name="coordIyyRaised" type="vtype"/>
        <xs:element name="coordIzzRaised" type="vtype"/>
        <xs:element name="coordIxyRaised" type="vtype"/>
        <xs:element name="coordIxzRaised" type="vtype"/>
        <xs:element name="coordIyzRaised" type="vtype"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:sequence>
      <xs:attribute name="uID" type="xs:integer" use="required"/>
      <xs:attribute name="configuration" type="xs:string" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="landingGearsType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
```

```
        <xs:element name="landingGear" type="landingGearType"
        maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="CELType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element name="controlEffector2" type="xs:int"/>
        <xs:element name="gain" type="xs:double"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="trimAnalysesType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element name="globalTrimDOFs" type="vtype"/>
        <xs:element maxOccurs="unbounded" name="trimCase"
        type="trimAnalysisType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="trimAnalysisType">
  <xs:complexContent>
    <xs:extension base="nd">
      <xs:sequence>
        <xs:element name="Mach" type="xs:double"/>
        <xs:element name="altitude" type="xs:double"/>
        <xs:element name="dynamicPressure" type="xs:double"/>
        <xs:element name="fixedVars" type="vtype"/>
        <xs:element name="fixedVals" type="vtype"/>
        <xs:element name="CSLink" type="tensorType"/>
        <xs:element name="constrainedDOFs" type="vtype"/>
        <xs:element name="supportedDOFs" type="vtype"/>
        <xs:element name="symmetry" type="xs:string"/>
        <xs:element name="controlEffectorLink" type="CELType"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:sequence>
      <xs:attribute name="uID" type="xs:integer" use="required"/>
    </xs:extension>
  </xs:complexContent>
```

```
</xs:complexType>
<xs:complexType name="flutterAnalysisType">
  <xs:complexContent>
    <xs:extension base="nd">
      <xs:sequence>
        <xs:element name="Mach" type="vtype"/>
        <xs:element name="altitude" type="xs:double"/>
        <xs:element name="dynamicPressure" type="xs:double"/>
        <xs:element name="reducedFrequencies" type="vtype"/>
        <xs:element name="velocityFactors" type="vtype"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:sequence>
      <xs:attribute name="uID" type="xs:integer" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="aerodynamicAnalysisType">
  <xs:complexContent>
    <xs:extension base="nd">
      <xs:sequence>
        <xs:element name="altitudes" type="vtype"/>
        <xs:element name="MachNumbers" type="vtype"/>
        <xs:element name="AoA" type="vtype"/>
        <xs:element name="cL" type="tensorType"/>
        <xs:element name="cD" type="tensorType"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:sequence>
      <xs:attribute name="uID" type="xs:integer" use="required"/>
    </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:complexType name="aerodynamicAnalysesType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="aerodynamicCase"
        type="aerodynamicAnalysisType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="aircraftAnalysesType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:all>
        <xs:element name="trimCases" type="trimAnalysesType"
        minOccurs="0" maxOccurs="1"/>
        <xs:element name="flutterCase" type="flutterAnalysisType"
        minOccurs="0" maxOccurs="1">
        <xs:element name="aerodynamicCases" type="aerodynamicAnalysesType"
        minOccurs="0" maxOccurs="1">
      </xs:all>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="performanceType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element name="maxCruiseMach" type="scalar"/>
        <xs:element name="cruiseAltitude" type="scalar"/>
        <xs:element name="maxRange" type="scalar"/>
        <xs:element name="serviceCeiling" type="scalar"/>
        <xs:element name="stallSpeedTakeoff" type="scalar"/>
        <xs:element name="stallSpeedlanding" type="scalar"/>
        <xs:element name="criticalFieldLength" type="scalar"/>
        <xs:element name="landingDistance" type="scalar"/>
        <xs:element name="cd0" type="scalar"/>
        <xs:element name="maxRCSValue" type="scalar"/>
        <xs:element name="flyoverNoiseFAR36" type="scalar"/>
        <xs:element name="sidelineNoiseFAR36" type="scalar"/>
        <xs:element name="thrustToWeight" type="scalar"/>
        <xs:element name="maxLoadFactor" type="scalar"/>
        <xs:element name="wingLoading" type="scalar"/>
        <xs:element name="diveSpeed" type="scalar"/>
        <xs:element name="maxRollRate" type="scalar"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:sequence>
      <xs:attribute name="uID" type="xs:integer" use="required"/>
      <xs:attribute name="configuration" type="xs:string" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="missionStageType">
  <xs:complexContent>
    <xs:extension base="nd">>
      <xs:sequence>
        <xs:element name="altitude" type="vtype"/>
        <xs:element name="Mach" type="vtype"/>
        <xs:element name="distance" type="scalar"/>
        <xs:element name="elapsedTime" type="scalar"/>
      </xs:sequence>
      <xs:attribute name="uID" type="xs:integer" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="airfoilsType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element name="airfoil" type="airfoilType"
        maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="missionType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element name="stage" type="missionStageType"
        maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="pointMassType">
  <xs:complexContent>
    <xs:extension base="nd">
      <xs:sequence>
        <xs:element name="mass" type="xs:double"/>
        <xs:element name="inertiaTensor" type="tensorType"/>
        <xs:element name="location" type="point"/>
        <xs:element name="provenance" type="provenanceType"/>
      </xs:sequence>
      <xs:attribute name="uID" type="xs:integer" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="pointMassesType">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="pointMass"
        type="pointMassType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
```

```
</xs:complexType>
<xs:complexType name="aircraftModelType">
  <xs:complexContent>
    <xs:extension base="nd">
      <xs:sequence>
        <xs:element name="airfoils" type="airfoilsType" minOccurs="0"/>
        <xs:element name="pointMasses" type="pointMassesType" minOccurs="0"/>
        <xs:element name="structuralElements"
        type="structuralElementsType" minOccurs="0"/>
        <xs:element name="wings" type="wingsType" minOccurs="0"/>
        <xs:element name="fuselages" type="fuselagesType" minOccurs="0"/>
        <xs:element name="propulsion" type="propulsionType" minOccurs="0"/>
        <xs:element name="landingGears" type="landingGearsType" minOccurs="0"/>
        <xs:element name="materials" type="materialsType" minOccurs="0"/>
        <xs:element name="performance" type="performanceType" minOccurs="0"/>
        <xs:element name="mission" type="missionType" minOccurs="0"/>
        <xs:element name="global" type="aircraftGlobalType"  minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="uID" type="xs:string" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name ="aircraft">
  <xs:complexContent>
    <xs:extension base="complexBaseType">
      <xs:sequence>
        <xs:element name="model" type="aircraftModelType"
        maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
</xs:schema>
```

Appendix B: Element Comparison of CPACS vs. ADML

CPACS elements (left column) and their matching (if any) ADML elements (right column). ADML elements (e.g., provenance) not available in CPACS are not listed.

CPACS	ADML
aircraft	aircraft
analyses	analyses
composites	compositeMaterials
costs	
enginePylons	
engines	engines
fuels	
fuselages	fuselages
global	global
landingGear	landingGears
materials	materials
model	model
payload	pointMass
profiles	airfoils
wings	wings
aeroelastics	
controlSurfacePolars	
dynamicAircraftModel	
landingGearPositionSafeMargines	
loadAnalysis	trimCases, flutterCases
massBreakDown	
paxFlow	
weightAndBalance	
dragStrut	
enginePylon	
fairing	
CPACS	ADML

frontAttachment	
frontAttachmentShackles	
frontPyramid	
fuselageAttachments	
innerSidePanel	
loadcarryingStructure	
lowerPanel	
outerSidePanel	
pins	
pylonBox	
pylonPins	
rearAttachment	
rearAttachmentShackles	
ribsDefinitions	
shackles	
struts	
tangentLinks	
upperLinks	
upperPanel	
wingAttachments	
engineMounts	
fan	
geometry	engineType(parametric engine geometry)
nacelle	
spinner	
cargoCrossBeams	
cargoCrossBeamStruts	
cargoDoors	
connections	
crashResults	
decks	

CPACS	ADML
floorPanels	
frames	
fuselageSkinGeometry	
fuselageSkinSegment	
hatracks	
paxCrossBeams	
paxCrossBeamStruts	
paxDoors	
positionings	
pressureBulkHeads	
rivetJointAreas	
seatRails	
seatRows	
sections	
segments	
stringer	
structure	
transformation	
windows	
xstruts	
wingBox	section (subelement of wing structure)
axles	
bogie	
cockpitControls	
commandCases	
controlDistributors	
controlLaws	
dragStrut	
mainactuator	
mainGear	landingGear (configuration=main)
mainStrut	

CPACS	ADML
noseGear	landingGear
pintleStrut	
piston	
sideStrut	
systems	
wheels	
actuators	
controlSurfaces	controlEffectors
leadingEdgeDevices	
positionings	location(attribute)
sections	sections
segments	
spoilers	
structure	structure
trailingEdgeDevices	
wing	wing
wingfuselageattachment	
wingFuelTank	