



A Mixed Integer Efficient Global Optimization Algorithm for the Simultaneous Aircraft Allocation-Mission-Design Problem

Satadru Roy^{*1}, Kenneth T. Moore^{†2}, John T. Hwang^{‡2}, Justin S. Gray^{§2}, William A. Crossley^{¶1} and Joaquim R. R. A. Martins^{||3}

¹*School of Aeronautics & Astronautics, Purdue University, West Lafayette, IN, 47907, USA*

²*NASA Glenn Research Center, Cleveland, OH, 44135, USA*

³*Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, 48109, USA*

Aircraft design optimization and airline allocation problems are two separate and well-researched disciplines, but very little literature exists that solved the design and allocation problems simultaneously. Among the limited number of related efforts that combine them, most follow a sequential decomposition strategy. This sequential strategy has been successful in addressing the combined large-scale problem but the approach does not capture the coupling that exists between the aircraft design and airline allocation disciplines. Solving the aircraft design and airline allocation as a monolithic problem makes it a Mixed Integer Non-Linear Programming problem which is very difficult to solve for large numbers of integer variables. Because no existing generalized MINLP solver can address this problem, this work proposes a new algorithm combining branch and bound, Efficient Global Optimization, Kriging Partial Least Squares, and gradient-based optimization to solve MINLP problems with 100's of integer design variables, 1000's of continuous design variables. The algorithm was applied to an 8 route coupled aircraft design and allocation problem with the 19 allocation variables and solving a 6000 variable aircraft design optimization problem using an Euler CFD simulation. This test problem provides several key challenges for a MINLP problem: a moderate integer design space, a large continuous design space, and expensive analysis models.

Nomenclature

EI	=	Expected Improvement
EV	=	Expected Violation
f_{local}^*	=	Optimal function value via local search
LBD	=	Global lower bound
M	=	Number of constraints that are functions of integer type variables
UBD	=	Global upper bound
\mathbf{x}_C	=	Vector of continuous type design variables
\mathbf{x}_I	=	Vector of integer type design variables
\mathbf{x}_I^0	=	Set of initial integer points
$\mathbf{x}_{\text{local}}^*$	=	Optimal solution via local search
$\mathbf{x}_{()}^{lb}, \mathbf{x}_{()}^{ub}$	=	Design variable bounds

*Ph.D. Candidate, AIAA Student Member

†Research Engineer, AIAA Member

‡Research Engineer, AIAA Member

§Research Engineer, AIAA member

¶Professor, AIAA Associate Fellow

||Professor, AIAA Associate Fellow

I. Introduction & Motivation

Aircraft design optimization is often posed as a nonlinear programming (NLP) problem, where the design variables are all continuous rather than mixed-discrete or mixed-integer type. Depending on the choice of fidelity in the various aircraft design disciplines (e.g., aerodynamics, structures, propulsion), the full multidisciplinary NLP can become very expensive computationally. On the other hand a simplified airline allocation problem, that assigns a fleet of aircraft to various routes in the network, can be formulated as a mixed-integer linear programming (MILP) problem. The integer design variables consist of the number of flights per day of a given type of aircraft on a given route. The continuous design variables consist of the numbers of passengers on those candidate route-aircraft pairings. Although aircraft design and fleet allocation problems can be solved separately, the two are inherently coupled; the design affects the performance of aircraft on the routes and hence the allocation, but the allocation also determines the utilization of the aircraft and in turn optimal design. Most studies that consider both the design and allocation problems adopt a sequential decomposition strategy.^{1,2,3} First the aircraft design optimization is performed and mission cost and performance data on each route for the new aircraft are obtained. Next a MIP problem is solved to find a fleet-level allocation that maximizes profit. However, sequential decomposition only partially captures the coupling that exists between the aircraft design and allocation, accounting for the affect of the design on the allocation problem but not vice versa. Previous work by the authors solved a fully coupled design-allocation problem with new monolithic formulation that solves both disciplines simultaneously by posing it as a NLP problem.^{4,5} In this approach, the aircraft shape and sizing design variables are optimized simultaneously with the airline allocation design variables, assuming the number of trips per aircraft type is continuous. Relaxing the integrality constraint on the allocation variables enables the use of gradient-based optimization with adjoint analytic derivatives and consequently also enables the aircraft design problem to include high fidelity analyses and make use of high performance computing resources. The key finding of this prior work was that profit could be increased by tailoring the design of the new aircraft to increase utilization for shorter range missions. In other words, it allowed an aircraft design optimization to be posed with an objective function of fleet-level profit. While the result clearly proves the value of considering the coupled problem, relaxing the integrality constraints makes the result less realistic. Additionally the NLP approach was shown to be sensitive to the initial starting guess for the relaxed allocation variable values. To avoid this issue, a starting point for allocation variables was computed from the sequential decomposition approach.

What is needed is an optimization algorithm that can handle the fully coupled design-allocation, respect the integrality of the allocation variables, be able to overcome poor initial guesses, accommodate a large continuous design space, and make efficient use of a parallel design space.

This led to the development of a new algorithm combining branch and bound, Efficient Global Optimization, Kriging Partial Least Squares and gradient-based optimization to solve design-allocation and airline allocation as an MINLP. A key feature of the new algorithm is its parallel scalability that enables it to work efficiently on problems that include a large NLP component which requires HPC resources. We named the new algorithm A Mixed Integer Efficient Global Optimization, AMIEGO. AMIEGO was implemented as an optimization driver in NASA's OpenMDAO framework, and can be run on an arbitrary MINLP. We first present results from running the new algorithm on simple test problems of varying scale to demonstrate its capability. Next we apply it to an 8 route design-allocation problem, run in parallel across 128 processors.

II. Literature Review

A. Low-fidelity aircraft design - allocation using Sequential decomposition approach

Among the early and existing studies that combine the aircraft design and allocation problems have done so within the context of system of systems.^{1,2,3,6,7,8} One of the first steps, was the MDO-motivated decomposition approach by Mane et al.,³ where they separated the MINLP problem into two smaller sub-space problems: aircraft design and airline allocation. The aircraft design problem sizes the new, 'yet-to-be-deployed' aircraft for a specified design mission range and payload via an NLP problem. The newly designed aircraft along with the existing set in the fleet are allocated to the route network via an MILP problem. Mane et al.³ also compare the decomposition approach with solving the coupled problem as an MINLP problem using algorithms like GA and Branch and Bound (BB). Although there were marginal differences in the objective function value using the different approaches, the decomposition approach showed potential

to obtain a quality solution with a reasonable cost.

B. High-fidelity aircraft design-allocation as large-scaled NLP problem

Coupling the design and allocation problems with a high level of detail necessitates mission analysis as part of the process. Given an aircraft design and a model for its aerodynamic performance, a mission analysis tool is needed to consider the full climb, cruise, and descent parts of the mission in computing its fuel burn and to determine the optimal altitude and Mach number profiles. A previous effort presented a modular mission analysis implementation⁹ within an adjoint based computational framework that facilitates integration among disciplines and computation of multidisciplinary derivatives. Based on this modular mission analysis tool, a study¹⁰ that solved the mission-allocation problem simultaneously led us in the direction of the current approach. The result found there for the simplistic 3-route airline network problem using the simultaneous approach provided a higher airline profit than the result from the decomposition strategy. However, this study did not yet include the design problem; the mission analysis and allocation problems were solved with a fixed aircraft design.

A follow-up effort⁵ extended the 3-route allocation-mission problem to consider a more realistic 128-route network by utilizing parallel computing, but with the integrality constraints relaxed. Each processor was assigned a single mission profile to analyze, and the mission profiles from each of the routes were optimized simultaneously as before, leading to over 6000 design variables. The parallel computational framework was an enabling method because it simplified the adjoint-based derivative computation across all of the routes, and handled the data passing among processors. Figure 2 shows that the allocation-mission optimization utilizes the long-range next-generation aircraft on the short-range routes more than the existing smaller aircraft. The next study⁴ added aircraft design variables to this 128-route problem. A computational fluid dynamics (CFD) solver was used simulate the aerodynamic performance of a next-generation aircraft design, and the wing shape was optimized along with the mission and allocation design variables. The results showed an increase in profit after AMD optimization, and the wing shape after AMD optimization was different from that resulting from a conventional design optimization. As with the previous study, this work relaxed the integrality constraint; the current study aims to address this simplification. Figure 1 shows a design structure matrix of the allocation-mission-design (AMD) optimization problem.

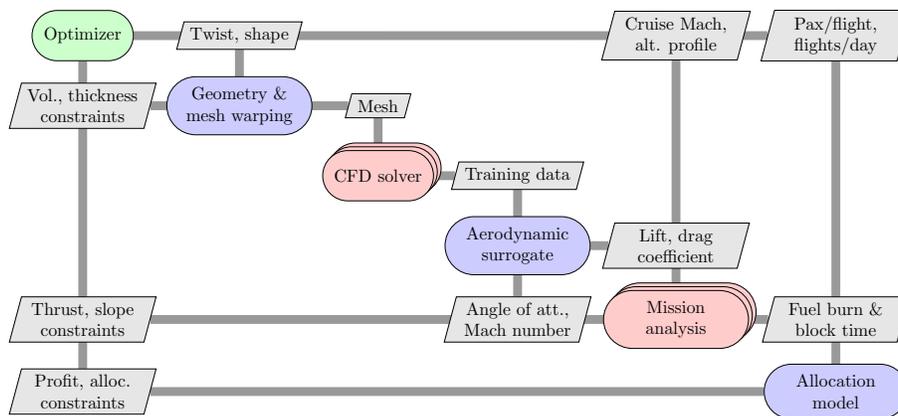


Figure 1: Design structure matrix for the allocation-mission-design optimization problem with the integer design variables relaxed.⁴

C. Low-fidelity aircraft design-allocation as small-scaled MINLP problem

With the inclusion of the integer type design variables the problem becomes very difficult to solve. Roy and Crossley¹¹ proposed a new MINLP approach to address this kind of problem. Keeping in mind the expensive nature of the high-fidelity aircraft design optimization, the approach uses an Efficient Global Optimization (EGO)-like framework to handle the integer and continuous type design variables using different optimizer. The framework leverage the global exploring capability of EGO to optimize the integer/discrete design space and uses the capability of gradient-based methods to efficiently explore the large scale continuous

design space. The paper demonstrates a small-scaled aircraft design- allocation problem posed as MINLP and solved using the EGO-like approach. The study used a simplistic Raymer-based¹² low-fidelity aircraft sizing tool to optimize the aircraft and obtain the aircraft performance data. The result using the EGO-like framework shows the simultaneous optimization yields slightly better aircraft design solution yielding better fleet-level profit compared to the sequential decomposition approach for the three-route problem.

The current effort combines these two works and leverages the computational framework of NASA's OpenMDAO^{13,14} to solve a high-fidelity Allocation-Mission-Design (AMD) optimization with integer constraint imposed for a moderately larger network problem. A detail description of the EGO-like framework together with newly added features to address large scale problem appears in the methodology section.

D. Surrogate Based Optimization

Engineering optimization problems are often limited by the number of the objective and constraint functions and the cost of evaluating these functions. The analyses needed to evaluate the functions are typically very expensive with long running time and, often, the routines providing these evaluations do not readily provide gradient information. These fit the definition of "black-box" functions.

One way to conduct optimization with black-box functions is via surrogate modelling. In this approach, actual evaluations of the black-box function at several discrete combinations of design variable values lead to a curve fit or other representation of the actual function. The resulting surrogate model can provide visualization, trade-off analysis and optimization. This lead to the development of surrogate based optimization algorithms, also known as Bayesian analysis algorithms¹⁵. One such surrogate based optimization algorithm that gained immense popularity over the years is Efficient Global Optimization (EGO)¹⁶. The algorithm starts with building a surrogate of the expensive black box function from an initial number of data points. The algorithm then uses the surrogate model to determine a location in the design space that provides the maximum likelihood of improving the objective function. A new black-box function evaluation at this point provides a new "infill" point. The surrogate model is then updated with this newly identified point and the steps are repeated until convergence. Initial development of EGO was for unconstrained continuous NLP problems. Later extensions of the algorithm can handle constraints^{17,18,19,20}. As the name suggest, the algorithm is effective in obtaining near-global solution for constrained expensive black box functions within a reasonable computational cost and has found its immense application in various fields. A very recent development in the field of surrogate based optimization for mixed integer type problems is by Muller.²¹ The proposed framework uses a similar approach of identifying an infill point that satisfies the integrality constraint, however our approach is more suited for applications with large number of design variables.

III. Methodology

To solve the aircraft design-allocation problem in a simultaneous approach posed as a MINLP problem, the section describes in details, the EGO-like optimization algorithm AMIEGO - A Mixed Integer Efficient Global Optimization. AMIEGO is the enhanced version of the EGO-like framework presented by Roy & Crossley¹¹ with added features to address large-scale MINLP or MDNLP problems. Although the algorithm can address both integer and discrete type design variables, we will use integer type variables to represent both integer/discrete type variables henceforth for the remaining of the paper. An overview of the AMIEGO algorithm appears in Fig.2 and a detail description of each step follows below. The red blocks use the EGO framework that explores the integer design space, while the blue block leverage the use of gradient-based approach to explore the large scale continuous design space.

Step 0: The algorithm first separates the integer type design variables from the continuous type design variables of the original MINLP problem.

Step 1: This step generates a set of initial integer points \mathbf{x}_I^0 . Any Design of Experiments (DOE) method can be used to generate the set of starting points. As traditional DOE methods do not generate integer points, the continuous results are rounded to meet the integrality criteria required in this step of the algorithm. This is not a hard requirement, but given the expensive nature of the objective function evaluation, we do not want to waste out any computational runtime of starting an expensive continuous optimization (step 2) starting from an continuous point.

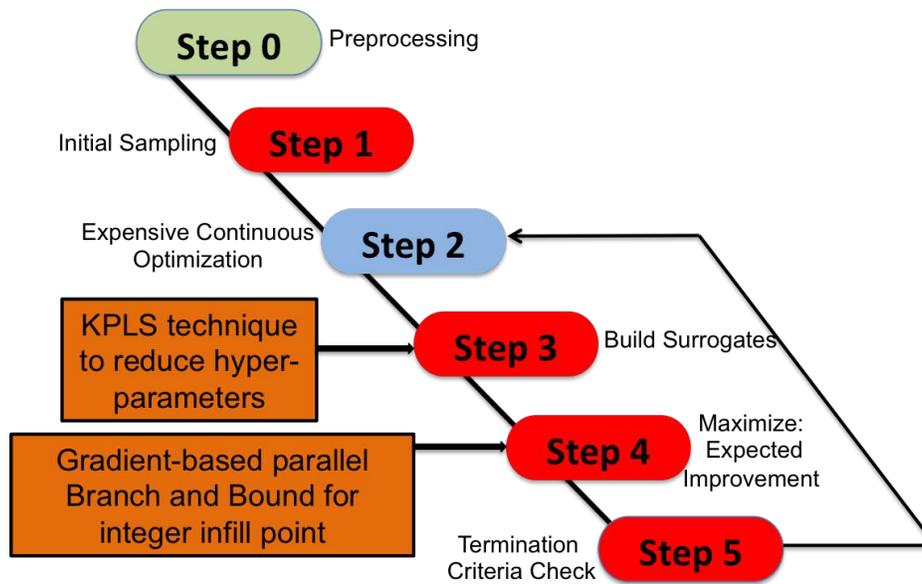


Figure 2: An overview of AMIEGO

Step 2: This step is different from the traditional EGO algorithm or any related work on Mixed Integer Surrogate Optimization. Instead of evaluating the objective/ constraints, the step performs a complete optimization with respect to the continuous design variables for each integer point using any gradient-based optimizer. All the continuous type design variables of the original problem appear in this step. The integer points from step 1 are supplied as parameters and stay constant during the optimization. The motivation to perform an optimization with respect to the continuous type design variables, is that, we expect this design space to be very large of the order of thousands of design variables^{4,5} and leverage the use of adjoint-based methods and parallel computing framework to solve such large scale design optimization problem in the continuous space. The subproblem formulation of this step appears below.

$$\begin{aligned}
 \text{Given: } \mathbf{x}_I, \quad & \underset{\mathbf{x}_C}{\text{Minimize:}} \quad f(\mathbf{x}_C, \mathbf{x}_I) \\
 \text{Subject to: } & g_1(\mathbf{x}_C) \leq 0 \\
 & g_2(\mathbf{x}_C, \mathbf{x}_I) \leq 0 \\
 & \mathbf{x}_C^{lb} \leq \mathbf{x}_C \leq \mathbf{x}_C^{ub}
 \end{aligned} \tag{1}$$

It is important to state that this optimization is still local with respect to the continuous design variables and is sensitive to the starting point as required by any gradient-based approach. For our allocation-design problem, we start-off from an existing aircraft data that we envision to model as an future aircraft. As an example, we may use an existing B737-800 aircraft as our starting point with the idea to obtain a better futuristic version of B737-800 considering the fleet-level objective.

Step 3: Surrogate models are built using the result of the continuous optimization obtained in the previous step. Any Gaussian based surrogate modelling technique may be used. We are using Kriging for training the surrogates. These surrogate models, as a function of the integer type design variables \mathbf{x}_I include:

- A surrogate model trained using the integer points at which the objective function is minimized with respect to the continuous design variables. This is the surrogate of the already minimized objective function, but with respect to the continuous design space.
- A surrogate model for each constraints that are function of the integer type design variables.

As we separate out the integer design variables from the continuous one, the surrogate building step of the algorithm sees only the integer type design variables of the original problem. However, as we seek to address large-scale design optimization problem, the number of design variables (hyper-parameters) involved in the

surrogate training process would be at least the same or more than the number of the decision variables. This is an NLP problem and may become computationally expensive as the number of hyper-parameters grow. The primary reason this optimization gets expensive is due to the presence of large covariance matrix that needs to be inverted several times during the optimization iterations. The number of times this matrix needs to be inverted is directly proportional to the number of hyper-parameters. Thereby, reducing the number of hyper-parameters would be one the strategies to reduce the computational cost associated with the covariance matrix inversion. To address this issue in AMIEGO, we combine Kriging with the Partial Least Square (PLS) regression as suggested by Bohlel et al.²² The paper proposes that combining Kriging with PLS regression, it is possible to significantly reduce the number of hyper-parameters i.e., the design-variables in the surrogate training optimization step. The method constructs a new covariance kernel function with reduced number of hyper-parameters based on the information provided by PLS technique. The most important property used in the approach to combine Kriging and the PLS regression is that the tensor product of covariance kernels is also a covariance kernel in the product space.^{22,23}

Step 4: This step finds an integer solution to the problem of maximizing the constrained expected improvement function. This auxiliary optimization problem to maximize the expected improvement function is also only with respect to the integer type design variables of the original problem. The expected improvement function accounts for both the uncertainty associated with the surrogate model and the improvement one can expect from the best known solution so far. As opposed to finding a continuous solution like in traditional EGO algorithm, this step finds a solution that also satisfies the integrality constraints. This integer solution in turn, is the solution to the integer design space and is the new infill point that appears as parameter to the continuous optimization in step 2. This leverages the global exploring capability of EGO algorithm to explore the integer design space in a computationally efficient manner. The problem formulation for this step appears below:

$$\begin{aligned} & \text{Maximize}_{\mathbf{x}_I} && \frac{EI(\mathbf{x}_I)}{1 + \sum_{i=1}^M EV_i(\mathbf{x}_I)/M} \\ & \text{Subject to} && \mathbf{x}_I^{lb} \leq \mathbf{x}_I \leq \mathbf{x}_I^{ub} \end{aligned} \quad (2)$$

The above formulation maximizes the constrained expected improvement function. Original formulation of EGO was for unconstrained problems. To handle the constraints, the approach uses the concept of expected violation similar to what is suggested in Ref^{17,19}. The expected violation concept is similar to the expected improvement approach, however the term $EV_i(\mathbf{x}_I)$ tends to zero when the i^{th} constraint is satisfied, otherwise it takes a positive value. Note, we calculate the expected violations of those constraints that are function of the integer type design variables only. From Equ.1, this would be $g2$ and not $g1$ as it is a function of \mathbf{x}_C .

Several optimization algorithms have difficulties maximizing the expected improvement function, due to its highly non-convex and multi-modal nature. Fewer algorithms are effective in finding the near global solution to this sub-problem. Most literatures uses evolutionary-based algorithms like Genetic Algorithm (GA) to maximize the expected improvement function. Further as we seek to obtain an integer solution, makes this sub-problem a Mixed Integer Non-Linear Programming (MINLP) problem. However, this auxiliary MINLP problem is computationally cheaper than the original MINLP of combined aircraft design and allocation problem. Also, for the objective 'Expected Improvement' function, after some rigorous mathematical simplifications, it is possible to obtain its gradient and the hessian information, which can be exploited by a mathematical programming solver for a global solution.¹⁶ To solve this auxiliary MINLP problem, we propose a new branch and bound (BB) approach that maximizes the constrained expected improvement function using a gradient-based optimizer.

A New Branch and Bound Algorithm using Gradient-Based approach: With the recent advancement in the field of mathematical programming and its efficacy to solve large-scale problems, the authors propose a new Branch and Bound (BB) that uses only gradient-based optimizer to maximize the expected improvement (EI) function. Like traditional branch and bound algorithm, this new approach also maintains a valid lower bound and a global upper bound at each node. To obtain a valid lower bound at each node, we follow the approach suggested by Jones et al.,¹⁶ that uses gradient-based approach to maximize the expected improvement function. However, in our case we also seek solution that satisfies the integrality constraint. To achieve this, we propose an approach that branches a node from the solution of the upper bound as oppose to the solution of the lower bound, typically followed in the traditional BB for MIP problems. This

is due to the strategy we use to obtain the lower bound of the EI function. Although we know what a valid lower bound is for a node, we do not know what is the solution x^* for that value of the lower bound. The solution to the upper bound may be obtained via a sampling approach or via a gradient-based search or a combination of both. An overview of the new BB algorithm appears below:

```

Initialize;
while ActiveSet  $\neq \emptyset$  and  $(UBD - LBD) \geq Tol$  do
    Perform random sampling - Round the point to the closest integer point - Pick the point with the
    best objective value - Set this as the best random sample;
    Solve EI problem of the current node via local search starting from the above point - Obtain
     $\mathbf{x}_{local}^*$ ,  $f_{local}^*$  rounded to the nearest integer;
    if  $f_{local}^* > best\ random\ sample$  then
        | Set  $f_{local}^*$ ,  $\mathbf{x}_{local}^*$  as best random sample;
    end
    if  $f_{local}^* \leq UBD$  then
        | Set UBD as  $f_{local}^*$ . This is the valid global upper bound;
    end
    Delete the parent node from the ActiveSet;
    for  $i = 1, 2$  do
        | Create a node with current lower bound and a new upper bound  $\mathbf{x}^{ub} = floor(\mathbf{x}_{local}^* + \delta)$ ;
        | Create a node with current upper bound and a new lower bound  $\mathbf{x}^{lb} = ceil(\mathbf{x}_{local}^* + \delta)$ ;
        | For child node  $i$ , obtain a valid lower bound  $LBD_i$  of the EI function;
        | Add child node  $i$  to the ActiveSet;
    end
    Delete any  $node_k$  from the ActiveSet with lower bound  $LBD_k \geq UBD$ ;
    Select current node as node with least lower bound value - Set it as global LBD;
end

```

Algorithm 1: Overview of the new Branch and Bound algorithm

Step 5: The last step terminates the algorithm when the expected improvement value falls below a certain percentage of the best known feasible solution. The current implementation of the framework uses a tolerance limit of 0.1% of the best found solution.

IV. Implementing AMIEGO in OpenMDAO

The AMIEGO algorithm was implemented in OpenMDAO using a hybrid driver approach. The outer driver is the main AMIEGO driver which encapsulates the loop containing steps 2 through 5 in Figure 2, and explicitly performs steps 3 and 5. The driver takes as input a list of initial integer-design-variable sample points, which are used to build the initial surrogate model. The driver also takes as input an optional list of objectives and list of constraint values corresponding to each of the sample points. This way, all of the pre-optimizations can be performed and saved before running AMIEGO. However, if only the design variables are given, then the continuous optimization will be solved for all of them before starting step 2.

A major design requirement for AMIEGO was that it should facilitate swapping the algorithm used in performing the continuous optimization (step 2) or in solving the MINLP (step 4) with other available algorithms, including some that already existed as drivers in OpenMDAO. This was accomplished by creating two slots in the AMIEGO driver: one called *cont_opt* which contains the continuous optimization, and one called *minlp* which contains the MINLP optimization of expected improvement. The AMIEGO driver takes care of feeding each of these sub-drivers the data it needs to correctly setup and solve the problem. The AMIEGO driver is given all of the problem's design variables and constraints, and it passes the continuous design variables and all of the constraints to the gradient optimizer, and the integer design variables and just the relevant constraints to the MINLP optimizer.

The default continuous optimizer is the SLSQP²⁴ optimizer from scipy, though all of the results presented here use SNOPT.²⁵ Both of these are gradient optimizers. The default MINLP optimizer implements the gradient-based Branch and Bound algorithm presented in Section III. While nearly any of OpenMDAO's optimizers can be used in the continuous optimization step, the MINLP optimizer needs to return a new design along with an expected improvement, so in practice, swapping in a new algorithm requires custom

design. The Branch and Bound driver implements the algorithm presented in Algorithm I. The expected improvement and the LBD problems are both solved using the SNOPT optimizer.

A. Parallel Implementation

Of all the steps in the AMIEGO algorithm, the Branch and Bound (step 4) present the most potential for performance gains from parallelization. Starting from a single initial candidate integer node, the algorithm generates two child nodes each iteration. Provided neither of them have yielded a lower bound that exceed the global UBD, we now have two nodes that can be evaluated independently. Algorithm 2 shows the parallel modifications made to Algorithm 1.

```

Initialize;
while ActiveSet  $\neq \emptyset$  and  $(UBD - LBD) \geq Tol$  do
  forall the  $N$  processors, take one of the current nodes if there are enough nodes in active set,
  otherwise idle do
    Perform random sampling - Round the point to the closest integer point - Pick the point with
    the best objective value - Set this as the best random sample;
    Solve EI problem of the current node via local search starting from the above point - Obtain
     $\mathbf{x}_{local}^*$ ;  $f_{local}^*$  rounded to the nearest integer;
    if  $f_{local}^* > best\ random\ sample$  then
      | Set  $f_{local}^*$ ,  $\mathbf{x}_{local}^*$  as best random sample;
    end
    if  $f_{local}^* \leq UBD$  then
      | Set UBD as  $f_{local}^*$ . This is the valid global upper bound;
    end
    Delete the parent node from the ActiveSet;
    for  $i = 1, 2$  do
      | Create a node with current lower bound and a new upper bound  $\mathbf{x}^{ub} = floor(\mathbf{x}_{local}^* + \delta)$ ;
      | Create a node with current upper bound and a new lower bound  $\mathbf{x}^{lb} = ceil(\mathbf{x}_{local}^* + \delta)$ ;
      | For child node  $i$ , obtain a valid lower bound  $LBD_i$  of the EI function;
      | Add child node  $i$  to the ActiveSet;
    end
  end
  Delete any  $node_k$  from the ActiveSet with lower bound  $LBD_k \geq UBD$ ;
  Select up to  $N$  current nodes as nodes with least lower bound values - Set the lowest as global
   $LBD$ ;
end

```

Algorithm 2: Overview of the new Parallel Branch and Bound algorithm

With the independence of the nodes in the active set, most of the algorithm can be done in parallel. Granted, this may cause some nodes to be executed in parallel that would have been pruned in serial, but overall the algorithm converges in less global iterations of branch and bound. A further modification called aggressive sampling can be made to the algorithm so that the active set is initially populated with a set of N candidate nodes, where N is the number of processors, so that processor utilization is maximized right from the first iteration. Figure 3 below demonstrates how aggressive splitting strategy allocates starting nodes to the 128 available processors for the high fidelity aircraft allocation-mission-design optimization problem (discussed in detail in Section VI).

V. Test Problems

A. 2D Test Problems

We tested AMIEGO on several test problems of varying scale and difficulty levels. We started with a 2-dimensional Branin²⁶ and Griewank²⁷ function. Griewank function is a challenging global optimization problem given its multi-modal nature as appears in Fig.4b. Although these are not MINLP problems, we posed them as MINLP problem by enforcing integrality constraint on one of the variables. The results demonstrate the ability of AMIEGO to obtain the global solution for both these problems.

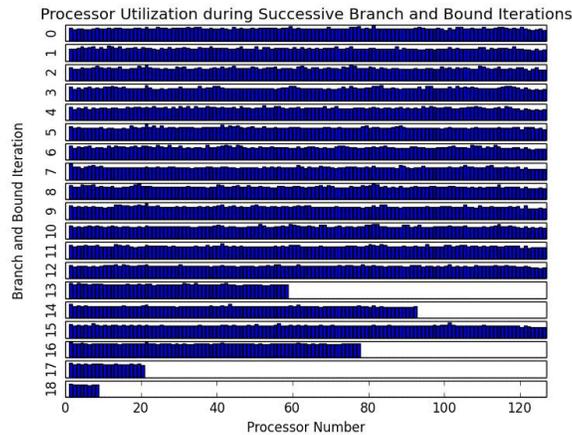


Figure 3: Processor utilization of the parallel branch and bound algorithm.

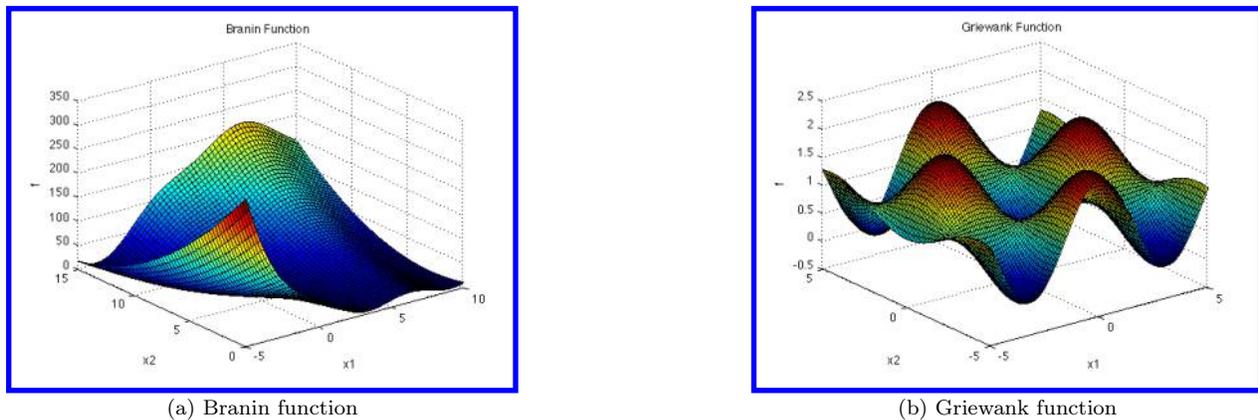


Figure 4: 2D test functions.

B. Engineering Problem with Discrete Variables

Next, we solved a three bar truss problem with three continuous type variables and three discrete material selection choices. The continuous design variables represent the cross-sectional area and the discrete variables corresponds to the material selection on each bar. The performance of AMIEGO seems promising (Table 1); being able to locate the global optima for the three bar truss is as few as 15 continuous optimizations, where complete enumeration would require 64 continuous optimization runs of each possible discrete material selection choice on the bars. The presence of discrete type design variables shows AMIEGO's ability to sort through various possible material selection choice in a computationally efficient manner. This is one of the most distinct abilities of AMIEGO over any state-of-the-art MINLP solvers.

Table 1: Comparison between complete enumeration and AMIEGO result for the three bar truss problem.

	AMIEGO	Complete enumeration
Area [cm^2]	[2.516, 2.038, 0.0]	[2.516, 2.038, 0.0]
Material	[3, 3, 4]	[3, 3, 1]
Mass [kg]	5.28	5.28
Total Continuous Optimization	15	64

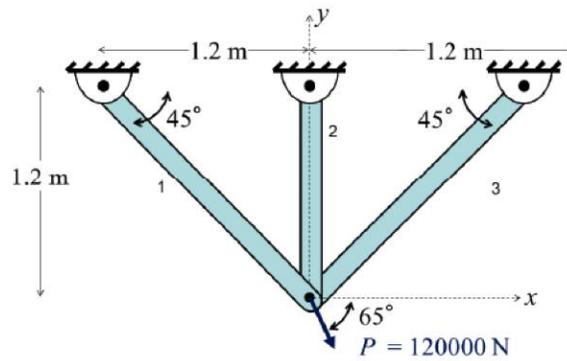


Figure 5: Three bar truss problem

C. Large-Scale Griewank Function

To demonstrate the scalability of AMIEGO algorithm and assess its performance for a large scale-problem, we also solved a 100-D Griewank function posed as MINLP problem with 50 integer type design variables and 50 continuous type design variables. We have also increased the domain of the design variables $\mathbf{x} \in [-600, 600]$. This makes the problem extra challenging given the vast possible enumeration of the integer design space. The parallel Branch and Bound algorithm in OpenMDAO leverages the use of 16 processors. Figure 6 shows the convergence plot of the 100-D Griewank function. Although, the global optima is at $f_{opt}^* = 0.0$, the algorithm shows promises of returning a near-global solution in as few as 130 continuous optimization runs, including the initial 100 sample runs.

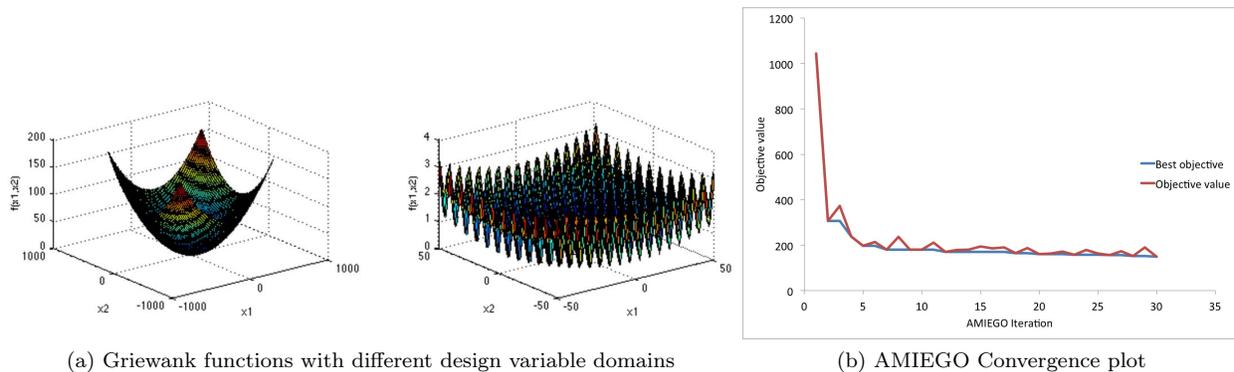


Figure 6: AMIEGO solving 100-D Griewank function posed as MINLP problem.

D. AMIEGO-FLOPS

Before integrating AMIEGO with high fidelity allocation-mission-design problem, we further tested out the performance of AMIEGO on an allocation-design problem using a medium fidelity, yet a credible sizing code - The Flight Optimization System (FLOPS)²⁸ developed by NASA. We picked a 11-route representative airline network with hub at Boston.³ The airline has two existing aircraft types already in operation - B757-200 and A320-200. The new 'yet-to-be-deployed' is a B737-800 sized aircraft that we seek to optimize for this airline network. As executing FLOPS to obtain a design solution is computationally cheap, we integrated AMIEGO-FLOPS within MATLAB²⁹ and use MATLAB's mixed-integer Genetic Algorithm (GA) to maximize the expected improvement function(step 4 of AMIEGO). Given the vast possible enumeration of the integer design space of which majority of them are infeasible, we restrict ourselves to a maximum of 6 flights per aircraft type per day. We note that increasing this limit delays the convergence of AMIEGO. This is expected, however, we believe a better scaling to the expected violation formulation might improve

the convergence performance and will be considered as part of the future work. A summary of the aircraft design variables appears in the Table 2.

Table 2: Summary of the aircraft design variables.

Aircraft design variables	Lower bound	Upper bound	Initial value	Optimal value
Aspect ratio	8.0	12.0	9.4	9.87
Taper ratio	0.1	0.5	0.159	0.275
Thickness to chord ratio	0.009	0.17	0.1338	0.099
Wing area [ft^2]	1000	2000	1345.5	1465.9
Wing sweep [deg]	0	40	25	21.31
Thrust (per engine) [lbs]	20000	30000	24200	24819

There are a total of 33 integer type variables of the allocation problem (11 route times 3 aircraft types). Figure 7 shows the AMIEGO-FLOPS results. We see there is definite change in the wing planform with an increased profit value in the final AMIEGO solution. We also compared the AMIEGO-FLOPS results with the sequential-decomposition approach of Mane et al.³ AMIEGO-FLOPS resulted in an \$23,847 increase in profit per day due to the monolithic approach of solving the design-allocation problem.

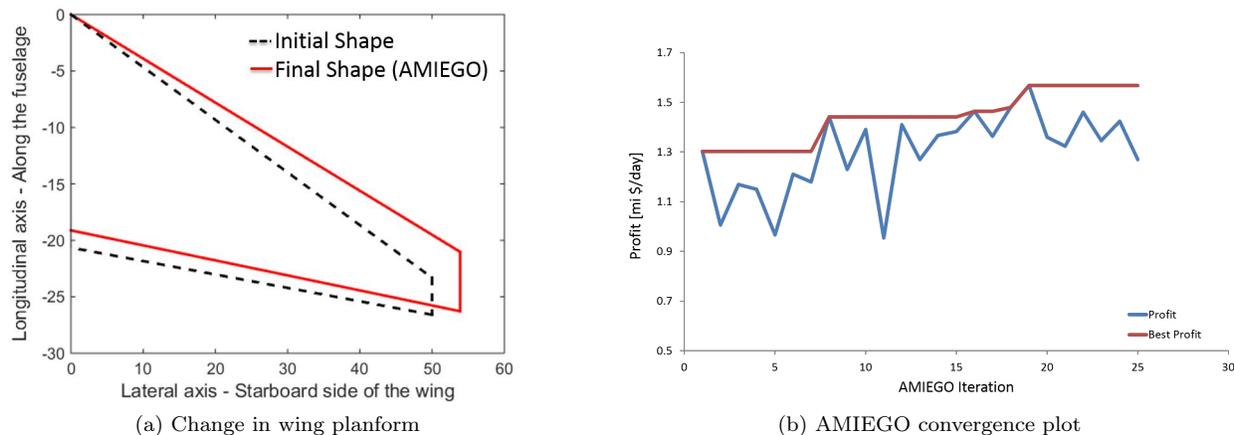


Figure 7: AMIEGO-FLOPS results.

VI. Results: AMIEGO-AMD

A. Background

The allocation-mission-design (AMD) optimization problem we solve in this paper is a modified version of the problem solved by Hwang and Martins,⁴ where we treat the allocation variables as integer instead of continuous variables, and consider an 8-route network rather than a 128-route network. The earlier paper used continuous relaxation, in essence treating all variables as continuous variables. However, the premise of this paper is that continuous relaxation leads to significant errors as the number of flights per day is $\mathcal{O}(1)$, meaning that rounding to the nearest integer represents relatively large changes to the design variables. Moreover, the rounded integer point may be infeasible as the rounding may violate constraints in the allocation problem.

The AMD problem combines the airline allocation problem, computational fluid dynamics (CFD), and mission profile design. AMD seeks to find the optimal design of a next-generation aircraft, while simultaneously determining its optimal utilization in a hypothetical airline network along with its optimal mission profiles. The perspective from design optimization is that optimization with CFD requires arbitrary selection of operating conditions—e.g., Mach number and lift coefficient at which to optimize a performance metric such as drag. New technologies such as hybrid electric aircraft concepts, continuous descent approach, and

morphing wings motivate modeling the entire mission profile—i.e., optimizing at the Mach numbers and lift coefficients seen during the entire mission profile. However, aircraft are flown on a wide variety of missions in practice, often on mission ranges that differ significantly from the design mission, so this motivates modeling the entire profile for not just one mission, but a set of candidate missions. Discretizing a set of mission profiles results in a prohibitively large number of operating conditions at which to perform CFD simulations, so the AMD approach is to use a surrogate model; however, since we also wish to vary the design of the aircraft during optimization, the surrogate model is dynamically retrained during the optimization iteration. Therefore, as illustrated in Fig. 1, each optimization iteration consists of four steps: the 16 parallel CFD analyses, the training of the surrogate model, the 128 parallel mission analyses, and the allocation analysis (computing profit and all relevant allocation constraints).

B. Model and optimization details

The use of the surrogate model instead of direct CFD evaluation makes the problem tractable, but the cost is still significant, so the AMD problem solved by Hwang and Martins⁴ is run on 128 processors. The 128 processors are split across 16 CFD evaluation points, with 8 processors per point, and during the mission analyses, each processor is assigned one mission because there are 128 missions in the airline network in the earlier work. The problem we solve here is structured the same way for CFD, but we solve an 8-route network as the starting step given the increase in difficulty presented by the integrality constraints. During the mission analysis stage, the first 8 processors analyze the 8 missions while the remaining processors are idling. Figure 8 shows the hypothetical 8-route airline network used in this study.



Figure 8: Hypothetical 8-route airline network (source: www.gcmap.com)

The objective function is profit, and the design variables include wing shape and twist variables (design), altitude and Mach number profile variables (mission), and variables representing the number of flights and passengers flown with a given aircraft type (allocation). We assume the airline has three types of aircraft: the Boeing 737-800, the Boeing 747-400, and a next-generation aircraft that is Boeing 777-sized. In the design part of the problem, there are 72 shape variables and 6 twist variables parametrizing the wing using B-splines and free-form deformation. The wing drag is modeled using a 3-D Euler solver with corrections for the viscous drag and the drag for the rest of the aircraft. In the mission part of the problem, there are $\mathcal{O}(10)$ B-spline control points parametrizing the altitude profile for each mission, with the number being dependent on the mission range. The equations of motion over the mission profile are discretized using a collocation strategy. Cruise Mach number is also a design variable for each mission, with a constant IAS climb assumed until the aircraft reaches the cruise Mach number. In the allocation part, the design variables are the number of flights per day and the number of passengers per day for each aircraft type and for each route. The numbers of flights per day are the integer variables.

Figure 9 shows the structure of the AMD optimization problem in relation to the AMIEGO solver. The AMIEGO solver represents the top-level loop. It calls the continuous optimizer each iteration with a specified flights per day array (step 2 of AMIEGO as appears in Fig. 2). The continuous optimizer, which is a gradient-based optimizer called SNOPT,²⁵ solves the AMD problem to maximize profit with flights per day fixed at the given values, but all other design variables allowed to vary. This value of profit is returned to the AMIEGO solver, which performs its necessary calculations (Fig. 2) and calls the continuous optimizer again with a new flights per day array (infill point) in the next iteration. For additional details on the AMD formulation, the reader is referred to Hwang and Martins.⁴

maximize	profit	
with respect to	flights per day design variables *(treated as integer)	
	passengers per flight design variables	(allocation:opt)
subject to	demand constraints	
	aircraft availability constraints	

B. RELAXED OPTIMIZATION APPROACH The relaxed optimization approach treats all design variables as continuous variables and solves the nonlinear programming (NLP) problem, (**relaxed:opt**).

maximize	profit	
with respect to	flights per day design variables	
	passengers per flight design variables	
	mission profile design variables	
	cruise Mach design variable	
	shape design variables	
	twist design variables	
subject to	demand constraints	(relaxed:opt)
	aircraft availability constraints	
	idle thrust constraint	
	maximum thrust constraint	
	maximum climb slope constraint	
	maximum descent slope constraint	
	geometric constraints	

C. AMIEGO APPROACH The AMIEGO approach solves the true, mixed-integer nonlinear programming (MINLP) problem with allocation, mission, and design all coupled, and solves the mixed-integer nonlinear programming (MINLP) problem, (**amiego:opt**). The only difference from (**relaxed:opt**) is that the flights per day design variables are treated as integer variables, as they should be.

maximize	profit	
with respect to	flights per day design variables *(treated as integer)	
	passengers per flight design variables	
	mission profile design variables	
	cruise Mach design variable	
	shape design variables	
	twist design variables	
subject to	demand constraints	(amiego:opt)
	aircraft availability constraints	
	idle thrust constraint	
	maximum thrust constraint	
	maximum climb slope constraint	
	maximum descent slope constraint	
	geometric constraints	

As part of the AMIEGO algorithm (step 2 of Fig.2), it formulates and solves NLP problems with the integer variables supplied as parameters, given by (**continuous:opt**).

given	flights per day variables *(integer inputs)	
maximize	profit	
with respect to	passengers per flight design variables	
	mission profile design variables	
	cruise Mach design variable	
	shape design variables	
	twist design variables	(continuous:opt)
subject to	demand constraints	
	idle thrust constraint	
	maximum thrust constraint	
	maximum climb slope constraint	
	maximum descent slope constraint	
	geometric constraints	

D. Results

This section presents the optimization results from the three approaches described in the previous section. We examine and compare the final profit value, accuracy, and the computational performance of each approach. To summarize, approach A is sequential optimization, where we perform multi-point aerodynamic shape optimization (design optimization); approach B is relaxed optimization where we solve the full AMD optimization ignoring the integrality constraints; and approach C is the AMIEGO algorithm, where we solve the full AMD optimization problem respecting the integrality constraints.

We note that the allocation model in the AMD implementation used for approach C has a units error that increases the optimized value of profit by about 10%. This error reduces the fuel burn for the two existing aircraft by a factor of 5 from their true values. The error was discovered after the AMIEGO results were generated, and we were not able to regenerate the results using the corrected fuel burn values because of the $\mathcal{O}(\text{day})$ -runtime, given the high fidelity nature of the AMD optimization. However, we generated the data for all of approaches A and B with this units mistake artificially added to be consistent across the approaches. The absolute values of the profits are impacted by this error, but we believe conclusions based on their relative values are still valid.

Figure 10 plots the convergence of the AMIEGO algorithm on the AMD MINLP problem. The initial data points are selected using Halton sampling, and the NLP problem, (continuous:opt), is solved at these flights per day values. The best profit from the initial Halton-sampled points is just over \$2 million per day. Immediately, the AMIEGO algorithm is able to improve the profit to over \$3 million per day, and eventually to \$3.350 million per day after 12 iterations. Here we have used a very simplistic ticket price model to arrive at a qualitative assessment of the approaches, ignoring the revenue management system and other sophisticated demand forecasting tools used by the airlines. These values of profit should not be viewed as an exact prediction of profit of an airline. The AMIEGO algorithm, in general, shows an improving trend with profit values getting better over the iterations. Computation times vary significantly for the (continuous:opt) problem, but they are typically on the order of several hours on a high-performance computing cluster with 128 processors, due to the high-fidelity nature of the analysis. However there were some issues with the algorithm as the continuous optimization failed to converge at some points. The convergence was hampered by CFD mesh issues due to negative volumes that led to the failure of the continuous optimization for some of the points chosen by AMIEGO as appears in Fig. 10 during iterations 8, 9, 10, and 12. We believe that this is a correctible issue that is localized to the CFD shape optimization part of the model, and it will be addressed in the future work.

Figure 11 shows the final profit values after optimization for approaches A, B, and C, normalized with respect to the sequential optimization approach (A), treated as the baseline case. The profits computed between the Sequential optimization (MILP) and AMIEGO algorithms are nearly identical. The relaxed optimization yields the highest profit. This is expected as this relaxed optimization approach starts off from the solution of the sequential optimization, and solves the combined problem by relaxing the integrality constraints on the allocation variables. However, the flights per day values from the relaxed optimization are not valid. Integer flights per day values are required since airlines operate on a daily schedule in part

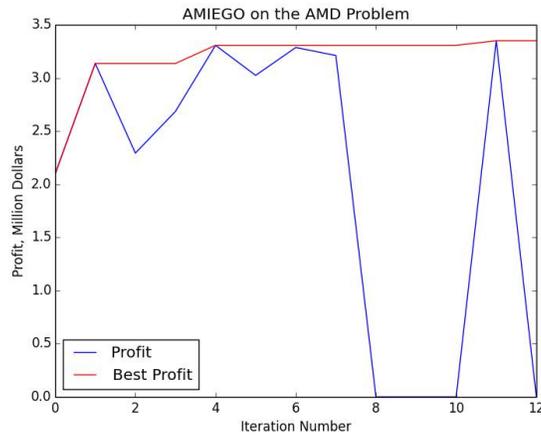


Figure 10: AMIEGO convergence plot for the AMD problem.

due to the fact that takeoffs and landings are not permitted during set hours at night. For example, if the integer relaxed optimization yields 2.7 flights per day on a given route, the 0.7 of a flight cannot carry over to the next day.

The fact that the AMIEGO approach yields a similar profit to the sequential optimization approach is unexpected; however, this is due to the CFD mesh issues that were previously discussed. The mesh issues prevent us from performing more iterations of the AMIEGO algorithm in Fig. 10 and cause premature termination after iteration number 12. Moreover, starting with more initial points would also improve the quality of the surrogate model, increasing the frequency with which points are found that increase profit. We expect that once the mesh issue and the lack of initial points are both addressed, the profit obtained by AMIEGO will be competitive with respect to the sequential optimization result.

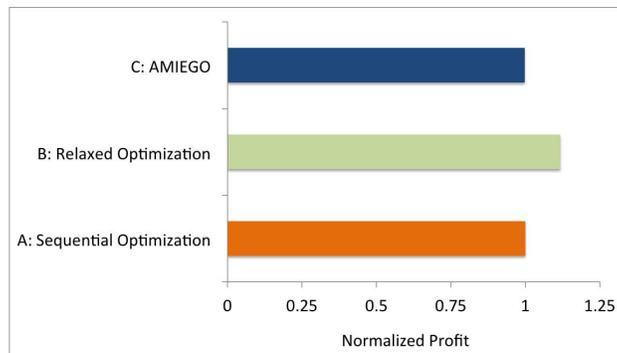


Figure 11: Comparison of the optimal profit (normalized) computed via the MILP, NLP, and AMIEGO algorithms.

A comparison of the allocation variables from the three methods appears in Figure 12. That data shows that the MILP and AMIEGO methods do have similar, but not identical allocations. The similarity in the profits between these allocations indicate that the final objective space is fairly flat.

VII. Conclusion & Future Work

With the intent to obtain near-global solution within limited computational budget, the framework seems appropriate for the high-fidelity allocation-mission-design problem. The ability of the AMIEGO to explore the integer type design space in a cost effective manner compliments with the efficiency of the gradient-based optimization algorithm to solve large scale design problems in the continuous space, all within NASA's OpenMDAO leveraging its parallel computing and computational framework. We have successfully

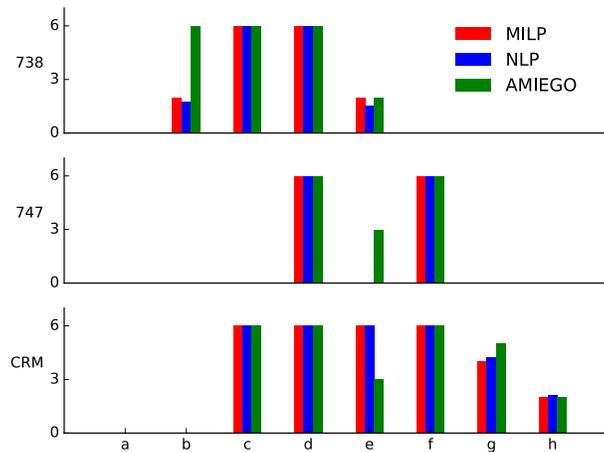


Figure 12: Comparison between the aircraft allocations computed using the MILP, NLP, and AMIEGO algorithms.

integrated the Kriging Partial Least Square method to our surrogate training model that drastically reduces the number of hyper-parameters needed to model the surrogates, thereby enabling us to address moderate scale integer design space problems. The gradient-based parallel Branch and Bound algorithm seems effective in obtaining a good integer infill point within reasonable computational time.

As a part of the future work, we would like to investigate in more details the CFD mesh issues leading to negative volume at the time of solving the high fidelity allocation-mission-design optimization. Next, we would like to solve a slightly larger airline network problem thereby increasing the number of integer type design variables in AMIEGO. Although we have demonstrated AMIEGO's ability to solve a 100D Griewank function problem with 50 integer variables, a similar scale high-fidelity allocation-mission-design will be something we would like to explore in the future. Further we can increase the complexity of the problem by introducing additional complex systems to better mimic an airline operations. For instance, a Revenue Management System (RMS) that predicts the optimal fare on a given route considering uncertain passenger demand, will definitely add fidelity to the existing allocation model.

VIII. Expected Significance

The expected significance of this work is to deliver a novel engineering design process that accounts for the fleet level objective at the time of designing the new 'yet-to-be-deployed' aircraft in a mix of existing fleet considering various disciplines in a tightly coupled manner. The conventional multi-point aircraft design optimization does not capture the true operational characteristics as how the aircraft is being used by the airline. The proposed framework optimizes the aircraft considering the operational behavior of the airline, as how they will use the newly designed aircraft along with the existing aircraft that will maximize the fleet level objective.

Acknowledgments

This work is supported by NASA through grant number NNX14AC73A.

References

¹Crossley, W. A., Mane, M., and Nusawardhana, A., "Variable Resource Allocation Using Multidisciplinary Optimization: Initial Investigations for System of Systems," *AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2004.

²Nusawardhana, *Dynamic Programming Methods for Concurrent Design and Dynamic Allocation of Vehicles Embedded in a System-of-systems*, Purdue University, West Lafayette, IN, 2007.

³Mane, M., Crossley, W. A., and Nusawardhana, A., "System-of-Systems Inspired Aircraft Sizing and Airline Resource Allocation via Decomposition," *Journal of Aircraft*, Vol. 44, No. 4, July 2007, pp. 1222–1235.

- ⁴Hwang, J. T. and Martins, J. R. R. A., "Allocation-mission-design optimization of next-generation aircraft using a parallel computational framework," *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, edited by A. I. of Aeronautics and Astronautics, January 2016.
- ⁵Hwang, J. T. and Martins, J. R. R. A., "Parallel allocation-mission optimization of a 128-route network," *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, June 2015.
- ⁶Davendralingam, N. and Crossley, W. A., "Robust Approach for Concurrent Aircraft Design and Airline Network Design," *Journal of Aircraft*, Vol. 51, No. 6, Nov. 2014, pp. 1773–1783.
- ⁷DeLaurentis, D. A., Crossley, W. A., and Mane, M., "Taxonomy to Guide Systems-of-Systems Decision-Making in Air Transportation Problems," *Journal of Aircraft*, Vol. 48, No. 3, May 2011, pp. 760–770.
- ⁸Mane, M. and Crossley, W. A., "Allocation and design of aircraft for on-demand air transportation with uncertain operations," *Journal of aircraft*, Vol. 49, No. 1, 2012, pp. 141–150.
- ⁹Kao, J. Y., Hwang, J. T., Martins, J. R. R. A., Gray, J. S., and Moore, K. T., "A modular adjoint approach to aircraft mission analysis and optimization," *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, January 2015.
- ¹⁰Hwang, J. T., Roy, S., Kao, J., Martins, J. R., and Crossley, W. A., "Simultaneous aircraft allocation and mission optimization using a modular adjoint approach," *SciTech, 53rd AIAA Aerospace Sciences Meeting*, 2015.
- ¹¹Roy, S. and Crossley, W. A., "An EGO-like Optimization Framework for Simultaneous Aircraft Design and Airline Allocation," *SciTech, 54th AIAA Aerospace Sciences Meeting*, 2016.
- ¹²Raymer, D. P., *Aircraft Design - A Conceptual Approach*, AIAA, Inc., 2012.
- ¹³Gray, J. S., Moore, K. T., and Naylor, B. A., "OpenMDAO: An Open-Source Framework for Multidisciplinary Analysis and Optimization," *13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Fort Worth, TX, AIAA, AIAA-2010-9101*, AIAA, Fort Worth, Texas, August 2010.
- ¹⁴Heath, C. M. and Gray, J. S., "OpenMDAO: Framework for Flexible Multidisciplinary Design, Analysis and Optimization Methods," *8th AIAA Multidisciplinary Design Optimization Specialist Conference (MDO)*, Honolulu, Hawaii, 2012, pp. 1–13.
- ¹⁵Forrester, A., Sobester, A., and Keane, A., *Engineering Design via Surrogate Modelling: A Practical Guide*, Wiley Publishing, 2008.
- ¹⁶Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, 1998, pp. 455–492.
- ¹⁷Audet, C., Dennis, J., Moore, D., Booker, A., and Frank, P., "Surrogate-model-based method for constrained optimization," *LACSI at Rice University, AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2000.
- ¹⁸Basudhar, A., Dribush, C., Lacaze, S., and Missoum, S., "Constrained efficient global optimization with support vector machines," *Structural and Multidisciplinary Optimization*, Vol. 46, 2012, pp. 201–221.
- ¹⁹Gengembr, E., Ladevie, B., Fudym, O., and Thuillier, A., "A Kriging constrained efficient global optimization approach applied to low-energy building design problems," *Inverse Problems in Science and Engineering*, Vol. 20, No. 7, 2012, pp. 1101–1114.
- ²⁰Tutum, C. C., Deb, K., and Baran, I., "Constrained Efficient Global Optimization for Pultrusion Process," Tech. Rep. COIN: 2014020.
- ²¹Muller, J., "MISO: Mixed-Integer Surrogate Optimization Framework," *Optimization and Engineering*, Vol. 17, No. 1, March 2016, pp. 177–203.
- ²²Bouhleh, M. A., Bartoli, N., and Otsmane, A., "Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimensional reduction," *Structural and Multidisciplinary Optimization*, Vol. 53, 2016, pp. 935–952.
- ²³Liem, R. and Martins, J. R. R. A., "Surrogate models and mixtures of experts in aerodynamics performance prediction for mission analysis," *15th AIAA/ISSMO multidisciplinary analysis and optimization conference*, AIAA, 2014.
- ²⁴Kraft, "A software package for sequential quadratic programming." Tech. rep., DLR German Aerospace Center — Institute for Flight Mechanics, Koln, Germany, 1998.
- ²⁵Gill, P. E., Murray, W., and Saunders, M. A., "An SQP algorithm for large-scale constrained optimization," *Society for Industrial and Applied Mathematics*, , No. 1.
- ²⁶Surjanovic, S. and Bingham, D., "Virtual Library of Simulation Experiments: Test Functions and Datasets," <http://www.sfu.ca/~ssurjano/branin.html>, [Online: accessed October 2015].
- ²⁷Surjanovic, S. and Bingham, D., "Virtual Library of Simulation Experiments: Test Functions and Datasets," <http://www.sfu.ca/~ssurjano/griewank.html>, [Online: accessed October 2015].
- ²⁸McCullers, L. A., *Flight Optimization System (FLOPS)*, ATK Space Division, NASA Langley Research Center, 8th ed., June 2010.
- ²⁹MATLAB, *R2014b (8.4.0.150421)*, The MathWorks Inc., Natick, Massachusetts, September 2014.

This article has been cited by:

1. Satadru Roy, William A. Crossley, Kenneth T. Moore, Justin S. Gray, Joaquim Martins. Next generation aircraft design considering airline operations and economics . [[Citation](#)] [[PDF](#)] [[PDF Plus](#)]
2. John T. Hwang. Reconfigurable model execution in the OpenMDAO framework . [[Citation](#)] [[PDF](#)] [[PDF Plus](#)]