AIAA SciTech Forum
7-11 January 2019, San Diego, California
AIAA Scitech 2019 Forum

A Mixed Integer Efficient Global Optimization Algorithm with Multiple Infill Strategy - Applied to a Wing Topology Optimization Problem

Satadru Roy* and William A. Crossley†
Purdue University, West Lafayette, IN, 47907

Bret K. Stanford‡
NASA Langley Research Center, Hampton, VA, 23681

Kenneth T. Moore§
DB Consulting Group, Inc., Cleveland, OH, 44135

Justin S. Gray¶
NASA Glenn Research Center, Cleveland, OH, 44135

With the advancement in high performance computing and numerical optimization techniques, engineering design optimization problems are becoming more complex, larger scale, higher fidelity, and computationally more demanding, requiring longer run times than ever before. There exists methodologies and techniques that can address some of these challenges but very few can address all, and most are limited in the extent that these concerns can be addressed. With the goal of addressing such challenging engineering problems, we developed a new optimization algorithm, named AMIEGO, that combines concepts from surrogate-based optimization approaches, gradient-based numerical methods, Partial Least Squares, evolutionary algorithms, and Branch-and-Bound, providing newer capabilities that were not previously perceived. The effort here builds upon this previously developed optimization algorithm to include multiple infill sampling capability that combines the concept of generalized expected improvement function, unsupervised learning, and multi-objective evolutionary technique. To demonstrate, AMIEGO with the multiple infill capability (called AMIEGO-MIMOS) solves a series of increasingly difficult engineering design optimization problems. The results reveal the performance of the new approach is problem dependent. When applied to a ten-bar truss problem, the newly proposed multiple infill strategy consistently leads to a better design solutions when compared to the existing CPTV method (implemented with the context of the AMIEGO algorithm). On the other hand, when applied to a mixed-integer high fidelity wing topology optimization problem - MIMOS, despite showing a steeper convergence at the start, eventually leads to an inferior solution as compared to CPTV approach. These results also reveal that a small number of starting points, in general, are sufficient to lead to a good overall solution.

Nomenclature

$AMIEGO$	=	A Mixed Integer Efficient Global Optimization
EGO	=	Efficient Global Optimization
EI	=	Expected improvement function
f	=	Objective function
f_{min}	=	Best objective function

*Post-Doctorate Associate, Aeronautics and Astronautics, 701 W. Stadium Ave., AIAA Senior Member
†Professor, Aeronautics and Astronautics, 701 W. Stadium Ave., AIAA Associate Fellow
‡Research Aerospace Engineer, Aeroelasticity Branch, 218 Dodd Blvd, AIAA Associate Fellow
§Research Engineer, AIAA Member
¶Research Engineer, AIAA Member

g	=	Parameter that drives the expected improvement function
h	=	Problem constraints
$MIMOS$	=	Multiple Infill via Multi-Objective Strategy
$MISO$	=	Mixed Integer Surrogate Optimization
PBS	=	Present Best Solution
s	=	Surrogate model error
x_C	=	Continuous type design variables
x_I	=	Integer type design variables
$\hat{y}$	=	Model prediction

I. Introduction and Motivation

With the advent of Multidisciplinary Design Optimization tools, it is now possible to combine a large number of disciplines within a single platform, thereby providing the tools and capabilities to address more complex, larger-scale engineering problems than ever before. However, as the problem complexity grows, the analysis tools needed to evaluate the objective and constraint functions become increasingly computationally expensive with long run times. Furthermore, depending on the complexity of the problem, they may also have large numbers of decision variables, some of which may be integer/discrete in nature. Such problems fall under the classification of a large-scale expensive Mixed Integer/Discrete Nonlinear Programming (MINLP/MDNLP) problem and are extremely difficult to solve. On the one hand, there exists numerous numerical optimization techniques that use gradient information to efficiently address large-scale problems that are purely continuous. With the advent of adjoint-based methods, it is now possible to efficiently compute the gradients and, hence, solve a very large-scale design optimization problem of the order of thousands of design variables [1, 2]. On the other hand, there exists several surrogate-based optimization strategies for computationally expensive analysis tools [3–8]

There are very few works that try to solve problems that involve both expensive analysis tools and a large number of mixed-integer/discrete type design variables. This is inherently a challenging problem due to the fact that it is difficult to build a surrogate and use the surrogate for reliable prediction when there are a large number of design variables involved, and it is equally challenging to perform an optimization on problems with expensive analysis tools and long run times. These problems are further aggravated due to the presence of mixed-integer or mixed-discrete type decision variables. This leads to some form of trade-off to either compromise the size (choose fewer decision variables) or the computational expense (reduce fidelity of the analysis tools) associated with the problem, when deciding on an optimization methodology. In this paper, we build upon a previously developed Mixed-Integer Nonlinear Programming (MINLP) solver, named AMIEGO [9–11], to address such large-scale expensive Mixed Integer Nonlinear Programming problems using a scheme with multiple sampling points per iteration, allowing one to perform expensive computer simulations in parallel.

AMIEGO uses a surrogate-based optimization approach based on an Efficient Global Optimization (EGO) approach [3]. The algorithm leverages the efficiency of a gradient-based optimizer to exploit the large-scale continuous design space together with the efficient global exploring capability of EGO to explore the integer/discrete type design space. In doing so, it combines the benefits that these individual optimization approaches provide, i.e., the ability to handle a very large-scale design optimization problem in the continuous space with expensive analysis models using a gradient-based approach and globally explores the integer/discrete design space using the EGO-like approach, limiting the expensive continuous optimization runs to as few as possible. This separation of the design variables or the bi-level programming approach, as commonly referred to in the literature [12], has several implicit benefits that were not previously exploited within the context of surrogate-based optimization techniques.

First and foremost, the separation of variables in AMIEGO lets the surrogate-based optimization subproblem see only the integer/discrete type design variables of the original problem and lets the gradient-based optimizer handle all of the continuous design variables of the problem. This drastically reduces the dimensionality of the problem as seen by the surrogate-based optimization part of the algorithm and, hence, helps partially address the "curse of dimensionality" associated with surrogate modeling [6]. Further, the reduced design space leads to reduced computational burden on the surrogate-based optimization subproblem and enables it to efficiently and effectively explore the integer/discrete design space. A complete enumeration of this integer/discrete design space may lead to large possible combinations that are computationally intractable. Furthermore, ensuring a minimum distance of at least one unit between the integer/discrete points in the surrogate-based approach (integrality constraints of the problem), removes the "ill-conditioning" issues frequently encountered at the time of building surrogates, as this prevents two points from getting too close to each other [6]. Although handling a large pure integer/discrete design space is still a concern due to, again, what is known as the "curse of dimensionality", AMIEGO partially addresses this challenge by incorporating Kriging with Partial Least Square (KPLS) [13]. Introducing KPLS at the time of building the surrogate reduces the computational burden to estimate the hyper-parameters. This addresses one of the major problems related to building a Kriging surrogate model for higher dimensional problems. Furthermore, Kriging and an expected improvement (EI)-based sampling strategy in AMIEGO go hand-in-hand, as an EI-based strategy exploits information supplied by the Kriging surrogate to identify the next infill point.

Secondly, because all the continuous design variables are addressed by an efficient gradient-based approach, this allows AMIEGO to effectively enforce all the problem constraints, which is also an important concern in the context of surrogate-based optimization. Furthermore, with the advent of adjoint-based methods, the gradient-based approach can handle very large-scale design problems in the continuous space.

All these features in AMIEGO make it a suitable MDNLP/MINLP solver to address problems that are complex (highly nonlinear in objective and constraint space), computationally expensive, large-scale and have mixed-integer/discrete type decision variables. The premise of this paper is to build upon the existing AMIEGO algorithm to retain these important features that AMIEGO provides and propose strategies that would extend its capability to sample multiple points per iteration for problems that will allow parallelization of the continuous portion of the problem. There have been several efforts to sample multiple infill points (i.e., points used to adaptively update the surrogate model) for surrogate-based optimization approaches and there is literature that point out the benefits of parallelization in this context [14]. The paper here proposes a novel multiple infill strategy that combines concepts from generalized expected improvement, machine learning, and multi-objective optimization and assess its performance on a difficult 10-bar truss engineering test problem. Finally, the parallel infill version of AMIEGO is applied to solve a wing topology optimization problem [15] and demonstrates its efficacy to solve such large-scale expensive Mixed-Integer Nonlinear Programming problems.

II. Literature Review

Recently, the idea of using surrogate-based optimization has received much interest for problems with expensive black-box type functions [3–7, 16]. The underlying idea to perform optimization using surrogate-based approaches employs an iterative way of identifying new infill points that updates the surrogate after each iteration. The key to obtaining a global solution using surrogate-based approaches lies in the balance between exploiting the surrogate (local search near the expected minima) and exploring the design space (where prediction error is expected to be high). One such surrogate-based optimization algorithm that has gained immense popularity ever since its introduction, is the Efficient Global Optimization (EGO) developed by Jones et al. [3].

The approach in this paper proposes to leverage EGO [3] as the underlying algorithm in a new approach that can address mixed variable (integer/discrete and continuous) large-scale problems in a computationally efficient manner. EGO, in its original format, cannot handle mixed discrete / integer variable problems. Like other surrogate-based optimization approaches, the EGO algorithm starts by building a surrogate model (Kriging) of the expensive black box function from an initial number of sampled data points. The algorithm then uses the surrogate model to determine a location in the design space that maximizes a merit function called “expected improvement” (EI). 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. The expected improvement function balances between the global exploration of the design space and the local exploitation of the region that is expected to have the minima. EGO’s initial development was focused on solving unconstrained, continuous NLP problems. Later extensions of the algorithm can handle constraints[17–19].

The above mentioned surrogate-based optimization approaches deal with pure continuous type design variables. However, the literature describing optimization using surrogate-based approaches for problems with mixed-integer type variables is very sparse. Lately, researchers have started looking for surrogate-based optimization approaches for mixed-integer type design variables [20–23]. One of the earliest applications of surrogate-based optimization for mixed-integer problems was by Holmstrom et al. [20]. The paper proposes a new adaptive radial basis function with a target sampling strategy and demonstrates its efficacy in solving low dimensional test problems (up to 11 dimensions). Davis and Ierapetritou [21] developed a methodology to address mixed-integer problems under uncertainty and with noisy data. The approach employs a Branch and Bound (BB) algorithm to enforce the integrality constraints and uses information from a Kriging model to explore each node of the BB algorithm.

Very recently, Muller et al. [22] developed a mixed-integer surrogate-based optimization tool SO-MI, capable of addressing both binary and integer type design variables. The method employs a stochastic sampling approach to obtain a new set of infill points and evaluates these points in parallel. The approach uses a cubic radial basis function (RBF) to build the surrogates. Later, MISO (a better version of SO-MI) was introduced by Muller [23]. The MISO framework employs several sampling strategies such as coordinate perturbation (CP), expected improvement (EI), random sampling (RS), target value (TV) to name a few. However, MISO framework, as represented in [23] are applicable only for unconstrained problems and may not able able to address very large-scale problems with thousands of design variables, as is the case with several complex engineering design problems. The AMIEGO algorithm, in a way, is comparable to the MISO-EI framework (expected improvement strategy). Both of them identify the new infill point by maximizing the expected improvement function. However, there are several differences between the two, some of which are already outlined in the introduction and motivation section. The major differences appear below.

One of the early motivations to develop the AMIEGO algorithm is to solve a very challenging engineering design problem [9–11, 24–26] that seeks to identify the best new aircraft design solution that maximizes a fleet-level objective considering airline operations and economics (see Fig. 1). The combined high-fidelity aircraft design, airline allocation and revenue management problem is expected to have a very large number of continuous design variables (thousands) and a moderate number of integer design variables (hundreds) for the allocation problem. To the authors best knowledge, no work exists that addresses problems with such a large number of design variables within a surrogate-based optimization approach due to the "curse of dimensionality" associated with surrogate modeling and it may not be possible to build a surrogate with that many design variables involved.

To address the mixed-discrete nature of the design variables, a bi-level or multi-level strategy may be followed where the top level uses an evolutionary or nature-inspired approach that drives the discrete/integer variables and the inner level uses gradient-based optimizer and deals with only the continuous design variables [27–30]. However for computationally expensive analysis tools that requires several hours of runtime and to meet the pressing desire to obtain a near-global solution under an extremely tight computational budget, an evolutionary-based approach may not yield a solution in practical time.

With rapid advancement in the field of numerical optimization and MDO frameworks to address large-scale design optimization problems, the authors seek to leverage this advancement and solve the large-scale design optimization problem in the continuous space using gradient-based approaches [31], adjoint methods [1, 2] and parallel computational frameworks [32, 33]. While only the integer type design variables are addressed using a surrogate-based optimization approach, the idea here is to use the AMIEGO algorithm to limit the expensive continuous optimization runs needed for the high-fidelity analysis to as few as possible, in contrast to limiting the number of actual, expensive function evaluations typically followed in the literature of surrogate-based optimization.

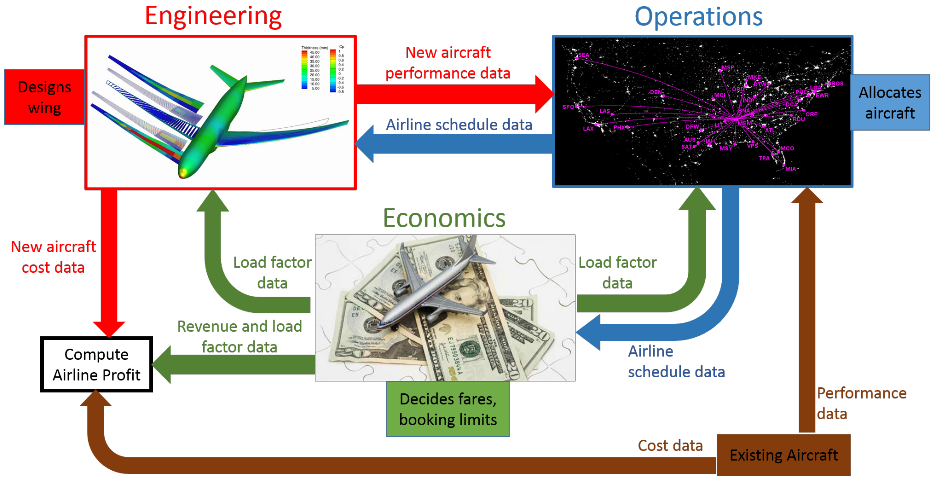


Fig. 1 A monolithic approach to simultaneously address engineering, operations and economics disciplines. (Adapted from Ref. [11])

Addressing the very large number of integer-type design variables that might be expected in some problems is still a challenge here. However, combining AMIEGO with Kriging Partial Least Squares [13] helps partially address this issue and extends the capability of AMIEGO with the intent to solve problems with about 100 integer/discrete type design variables as reported in Ref. [13]. The original version of AMIEGO samples only one infill point per iteration and all the parallel computing resources are devoted for the expensive continuous optimization. However, if CPU resources permit, one may infill multiple samples per iteration if expensive computer simulations may be evaluated in parallel independent of each other. The effort here demonstrates the multiple infill version of AMIEGO by solving a series of increasingly difficult engineering design optimization problem.

The concept of parallel infill sampling is not new. Haftka et al. [14] recently published a survey article comparing how different algorithms, leveraging parallelization, balance exploration and exploitation within a surrogate-assisted global optimization approach. The article seeks to point out the scope of additional improvements in the field of surrogate-based optimization that leverages parallelization. This motivated the authors to study and understand several existing multiple sampling strategies; the work here presents a new multiple infill strategy that combines concepts from generalized expected improvement, multi-objective evolutionary algorithms, and unsupervised machine learning, as presented in the following section.

III. Methodology

A. Original AMIEGO Algorithm

An overview of the original AMIEGO algorithm appears in Fig.2 and a brief description of each step (also outlined in Ref. [9–11]) follows below. The red blocks use the EGO-like approach that explores the integer design space, while the blue block leverages the use of gradient-based approaches to explore the large-scale continuous design space.

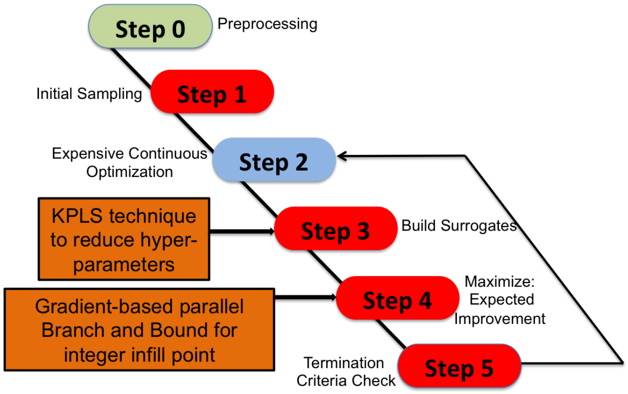

Step 0
Preprocessing
Initial Sampling
Step 1
Expensive Continuous Optimization
Step 2
KPLS technique to reduce hyper-parameters
Step 3
Build Surrogates
Gradient-based parallel Branch and Bound for integer infill point
Step 4
Maximize: Expected Improvement
Termination Criteria Check
Step 5

Fig. 2 An overview of the AMIEGO optimization algorithm.

- **Step 0:** The algorithm first separates the integer type design variables from the continuous type design variables of the original MINLP problem. Doing this enables the algorithm to handle a very large number of continuous design variables via adjoint-based numerical optimization techniques.

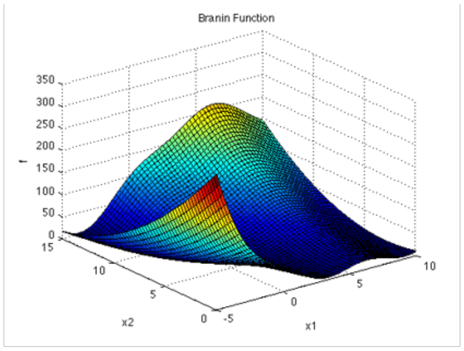
- **Step 1:** This step generates a set of initial integer points $\mathbf{x}_{\mathbf{I}}^{\mathbf{0}}$. Any Design of Experiments (DOE) method may be used to generate the set of starting points that also satisfies the integer constraints.

- **Step 2:** This step is different from the traditional original EGO algorithm or any related work on Mixed-Integer Surrogate-based Optimization. Instead of evaluating the objective/ constraints, the step performs a complete optimization with respect to the continuous design variables using a gradient-based optimizer, for each integer point. All the continuous type design variables of the original problem appear in this step as the decision variable of the inner-loop optimization. The integer points from step 1 are supplied as parameters and stay constant during the optimization. This step may be seen as a variant of bi-level programming, however the novelty of this approach is through leveraging a surrogate-based approach to address the integer/discrete type design variables as compared to traditional evolutionary approaches typically followed in the literature. Splitting the variables helps reduce the computational burden on the surrogate-based optimization part of the algorithm that deals with only integer/discrete type design variables of the original problem. The subproblem formulation of this step appears below.

$$\begin{aligned} \text{Given: } \mathbf{x_I}, \quad \underset{\mathbf{x_C}}{\text{Minimize:}}\ & f(\mathbf{x_C}, \mathbf{x_I}) \\ \text{Subject to: } & \mathbf{h}(\mathbf{x_C}, \mathbf{x_I}) \leq 0 \\ & \mathbf{x_C}^{lb} \leq \mathbf{x_C} \leq \mathbf{x_C}^{ub} \end{aligned} \tag{1}$$

It may be possible that for some integer points, the continuous optimization may not return a feasible solution (integer infeasible). In such cases, AMIEGO applies a penalty to the final objective value returned from the continuous optimization subproblem [26]. It is also important to state that this optimization is still local with respect to the continuous design variables and is sensitive to the starting point, as is the case with any gradient-based approach. For engineering design optimization problems, we recommend initializing the continuous design variables to an existing baseline design that the researchers envision to improve upon.

As an example, we showcase the 2-D Branin function (see Fig. 3) and demonstrate how splitting the design space could lead to two different subproblems. To pose this simplistic problem as a mixed-integer problem, we treat x_1 as the integer type design variable, while x_2 as the continuous type design variable.



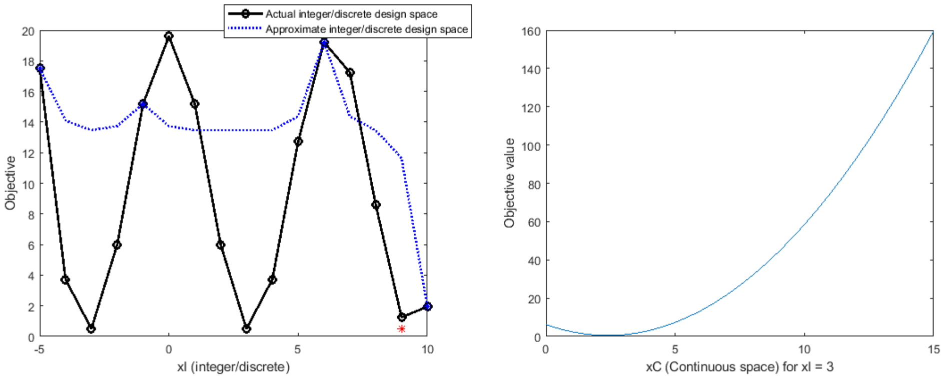
Global Optima: f(x*) = 0.494, x* = {-3, 11.94}, {3, 2.39}

Fig. 3 2-D Branin function posed as a mixed-integer nonlinear programming problem with x_1 considered as integer type and x_2 considered as continuous type.

Figure 4 shows the two subproblems that AMIEGO solves - one on the integer and the other on the continuous design space. These subproblems appear very different from the original 2-D Branin function. The EGO-like surrogate approach solves the wavy integer design space (figure to the left). Note each point in the integer design space (as denoted by the black circle) represents the final output from the continuous optimization subproblem in step 2. The blue dotted line shows the Kriging surrogate fit passing through the initial integer points at x_I = [-5, -1, 6, 10]. The red star at the bottom right shows the location of the next infill point as found by solving the EI maximization subproblem (using a single infill point criteria). It is important to note that for some problems, where x_I are allowed to take only discrete choices, no information actually exists between the points. However, within the context of AMIEGO we assumed there is continuity between the points and treat it as a continuous design space at the time of solving the EI maximization problem. This is a valid assumption as there exists no true order between a set of discrete choices and based on how we set the order, it will only change the discrete design landscape making AMIEGO think of it as a new problem [11]. The figure to the right shows the continuous design space of the original problem for a given integer input (here x_I = 3).

- **Step 3:** A surrogate model is built using the result of the continuous optimization obtained in the previous step. We use Kriging to train the surrogate. This surrogate model is a function of the integer type design variables $\mathbf{x_I}$ with the output response as the optimized objective function of the original problem, but with respect to only the continuous type design variables.

  One of the main drawbacks of the Kriging approach is that it becomes computationally very expensive to evaluate the hyper-parameters as the problem size scales. The number of hyper-parameters that need to be estimated at the time of building the surrogate model is the same as the number of design variables of the original problem (in our case, this is the number of integer/discrete design variables of the original problem). This estimation of



(a) Integer design space as seen by the EGO-like approach (b) Continuous design space as seen by the gradient-based optimizer

Fig. 4 AMIEGO solves two different subproblems for the 2-D Branin function

the hyper-parameters requires the user to solve for a separate optimization problem either using a global search technique such as a genetic algorithm or employing a multistart approach with a gradient-based approach. To overcome this challenge, AMIEGO combines Kriging with the Partial Least Square (PLS) regression as suggested by Bohlel et al.[13]. 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 a reduced number of hyper-parameters based on the information provided by the PLS technique. This feature in AMIEGO helps keep the computational cost associated with building the surrogate nearly independent of the number of integer design variables of the original problem.

- **Step 4:** This step finds an integer solution of the original problem by maximizing the expected improvement function. As opposed to finding a continuous solution as in the traditional EGO algorithm [3], 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 a parameter to the continuous optimization in step 2. The problem formulation for this step appears below:

$$\begin{aligned} \underset{\mathbf{x_I}}{\text{Maximize}} \quad & EI(\mathbf{x_I}) \\ \text{Subject to} \quad & \mathbf{x_I}^{lb} \leq \mathbf{x_I} \leq \mathbf{x_I}^{ub} \end{aligned} \tag{2}$$

Several optimization algorithms have difficulties maximizing the expected improvement (EI) function, due to its highly nonconvex and multimodal nature. Fewer algorithms are effective in finding the near-global solution to this subproblem. Most of the literature uses evolutionary-based algorithms like a genetic algorithm to maximize the expected improvement function. Furthermore, as we seek to obtain an integer solution, the subproblem becomes a Mixed Integer Non-Linear Programming (MINLP) problem. However, this auxiliary MINLP problem is computationally cheaper than the original expensive MINLP problem, given it works on the surrogate built in the previous step. Also, for the objective 'Expected Improvement' function, after some 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 [3]. AMIEGO provides the user with the option to use a GA or the newly developed gradient-based Branch-and-Bound approach available within OpenMDAO [32] to solve this EI maximization problem.

As mentioned earlier, the original version of AMIEGO identifies only one infill point per iteration, designed to cater to problems that require high performance parallel computing in solving the inner-loop continuous optimization in step 2. However, for problems that do not require parallel computing capabilities to solve a single continuous optimization problem, it may be desirable to sample multiple integer/discrete points and perform the continuous optimization in step 2 of the algorithm in parallel. The premise of this paper is to introduce this new capability in the AMIEGO algorithm, and let the users have the option to decide how many infill points they desire per AMIEGO iteration. The details about the new multiple infill sampling criteria appears in the subsequent section.

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

The new multiple infill strategy: AMIEGO-MIMOS

The traditional expected improvement (EI) function has two terms that balance between exploration and exploitation of the design space. However, for large scale problems, when it is not practical to sample many design points – given the vastness of the design space – the performance of the EI criteria may deteriorate [34], as it may lead to more exploration than exploitation. Sasena [8] proposed the concept of a generalized expected improvement function, where exploration and exploitation can be controlled using a parameter g. The generalized EI function, as shown in Eq. 3, uses a positive integer parameter g whose value may typically range from 0 to 20. Higher values of g tend toward global exploration while smaller values of g lead to more local exploitation. Sasena proposed this concept of generalized expected improvement within the context of a single infill strategy using a cooling criteria approach. The approach starts with a very high value of g (g=20) for global exploration and then as the iterations progress, it reduces the value in steps. Finally at around 15 iterations, it sets the value of g to zero, thereby leading to a full local exploitation around the best found solution so far.

$$\begin{aligned}
&E(I^g) = s^g \sum_{k=0}^{g} (-1)^k \left(\frac{g!}{k!(g-k)!} \right) z^{(g-k)} T_k \\
&\text{where,} \\
&\quad z = \frac{f_{min} - \hat{y}}{\hat{\sigma}} \\
&\quad T_k = -\phi(z) z^{k-1} + (k-1) T_{k-2} \quad \forall k \geq 2 \quad \text{and} \\
&\quad T_0 = \Phi(z), \quad T_1 = -\phi(z)
\end{aligned} \tag{3}$$

In this paper, we seek to leverage this concept of generalized expected improvement for multiple infill points, but within the context of a multi-objective approach. The strategy seeks to uniformly sample across three different pools of candidate points, generated by solving a multi-objective (MO) optimization problem, hence the name AMIEGO-MIMOS (AMIEGO-Multiple Infill via a Multi-Objective Strategy). This multi-objective formulation seeks to simultaneously maximize three objective functions: 1. the original expected improvement function (balance between exploration and exploitation), 2. the distance from an existing evaluated point (pure exploration), and 3. the generalized expected improvement function for g=0 (pure exploitation). We employ a binary-coded multi-objective evolutionary algorithm [35] that also satisfies the integer constraints of the design variables and generates a 3-D Pareto front of non-dominated designs. This idea is similar to Ref. [36] that uses a 2-D Pareto front by maximizing only the first two objectives i.e. the original EI and the design space exploration. The idea here extends to add another objective to the MO problem formulation for a better local exploitation. Furthermore, this idea also employs a novel sampling strategy that helps decide how to select points from the Pareto front.

Once we have a set of non-dominated designs, the next task is to sample points from this 3-D Pareto front. We employ an unsupervised machine learning technique to cluster the points based on their spatial location in the objective space. We use the k-mean clustering algorithm to generate as many clusters as the number of desired sample points per AMIEGO iterations. Figure 5 shows an example 3D non-dominated set of designs classified into 6 different-colored clusters and the corresponding cluster centroids, represented as stars. Figure 6 shows the 2D projection of the Pareto front across each objective pair. Note, the center plot (objective 1: balance and objective 3: exploit) shows a "dominating" trend of designs, where an improvement across one objective also improves the other objective, but at the cost deteriorating the objective 2 (explore). The plot at the center also shows a small region of non-dominated designs between the objective 1 and 3 towards the left bottom corner.

Next, we rank and sort these clusters based on the location of their centroid information – from best ranked to worst ranked clusters across each objective. Starting from objective 1, we then select the best ranked cluster and sample a point from this cluster (every cluster has a group of points) and then make this cluster unavailable for selection for the other objectives. We repeat this process for the remaining two objectives. After the end of this one round of sampling, we have three sample points in our database. We continue the above steps until we obtain the required number of sample points.

This section describes the approach we employ to select a point from a specific cluster. For objective 1 (balance:

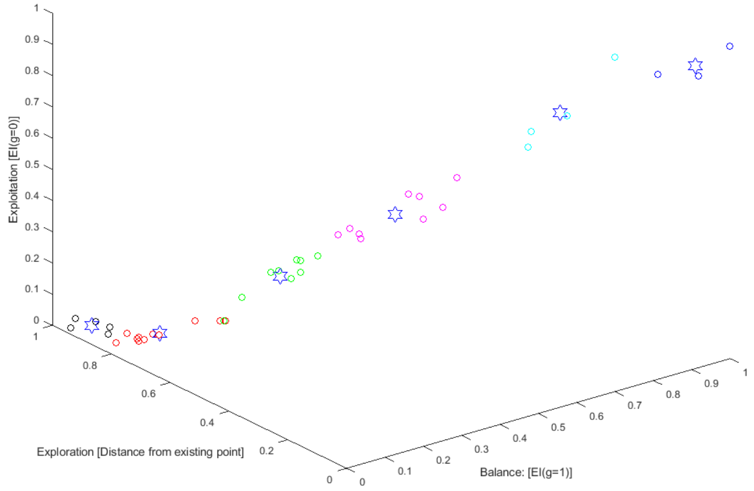


Fig. 5 3-D Pareto front showing the non-dominated designs, classified into 6 different clusters, and the location of the centroid of each of the clusters (represented by the stars)

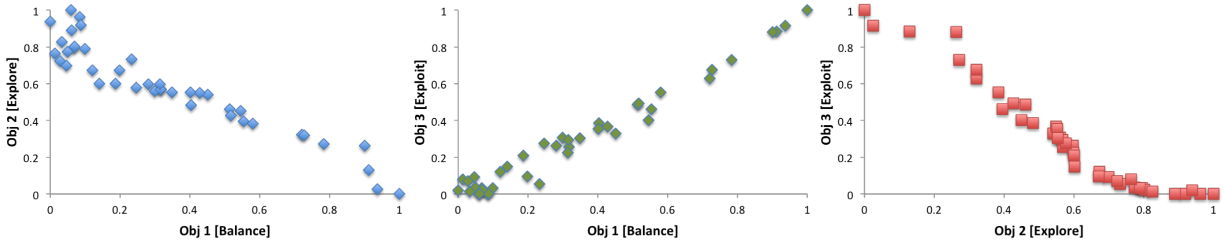


Fig. 6 2-D projection of the 3-D Pareto front across each objective pair

original expected improvement (g=1 in Eq. 3)), we select the point that has the lowest EI value. For objective 2 (exploration), we select the point that is farthest from its closest existing point. For the third objective (exploitation: expected improvement (g=0 in Eq. 3)), we select a point within the cluster that is nearest to the present best solution.

As a basis for comparison, we also implement an existing infill strategy, previously developed for single sample per iteration problems, and extend its capability to sample multiple infill points. We use the CPTV approach that uses a coordinate perturbation (CP) approach with a targeted value (TV) search. A detailed description of the approach appears in Ref. [23]. However, unlike its implementation in the original form, we use this infill strategy in the step 4 of the AMIEGO algorithm and call this AMIEGO-CPTV. The fundamental difference between the original implementation of CPTV and the current implementation is that as AMIEGO separates the variables, CPTV works only on the integer/discrete design space and provides these points as parameters to the inner continuous optimization in step 2.

IV. Application to engineering design optimization problems

The new multiple infill strategy is applied to two engineering design problems of increasing complexity - a ten bar truss and a wing topology design optimization problem.

A. Application I: Ten-bar Engineering Truss Problem

The ten bar truss problem is an example of a mixed-discrete nonlinear programming problem due to the presence of both categorically discrete (non-ordered) and continuous type design variables along with nonlinear objective and constraints. Figure 7 shows the ten-bar truss problem that has ten continuous design variables signifying the cross sectional area of each bar and has ten discrete type variables corresponding to material selection for each bar from the available set of choices from Aluminum, Steel, Titanium and Nickel. The goal is to reduce the total weight of the 10-bar truss system and the problem constraints limit the maximum allowable stress on each bar to be within the material allowable yield stress.

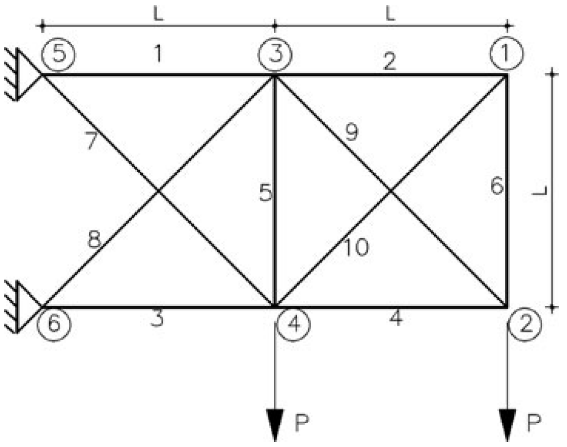


Fig. 7 Engineering 10-bar truss problem

To test the performance of multiple infill strategies on the ten bar truss problem, we present the following test cases - optimization dependence on different number of samples per iterations, and different number of starting points.

1. Effect on different number of samples per iteration of AMIEGO

To assess the performance of the AMIEGO-MIMOS algorithm, we first conducted an experiment with 31 initial samples and vary the number of infill points per iteration. Figure 8 shows the ten bar truss for different numbers of infill points per iteration using the AMIEGO-MIMOS strategy. It has four different cases (represented by four different colors in the plot), each represents a different number of samples per iteration. Case 1 (blue line) samples only one point per iteration. Referring back to the description of MIMOS, this case only samples one point from the clusters that is ranked highest in objective 1 (expected improvement function in Eq 3 with g=1). This essentially makes the algorithm revert back to the classic AMIEGO approach [9] with a single infill strategy using the traditional EI criteria. The remaining three cases follows the sampling strategy as discussed in the previous section.

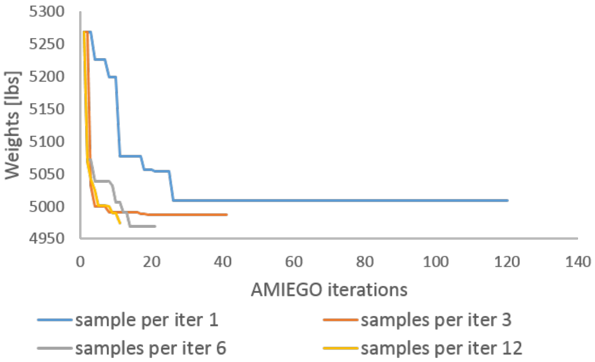
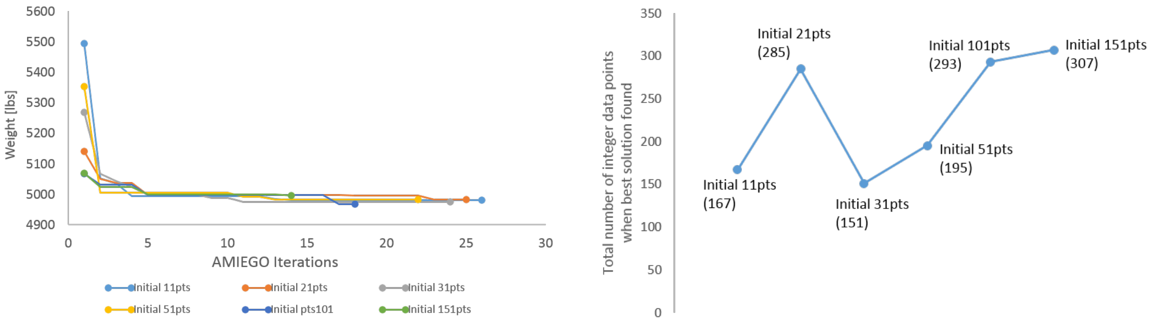


Fig. 8 Convergence plot: AMIEGO-MIMOS for different samples per iteration

It is intuitive to see that Case 1 takes longer to converge as compared to other cases. This is expected, because in Case 1 we are sampling only one point per AMIEGO iteration, whereas we are sampling multiple points in other cases. However, they all undertook the same number of data (discrete) point evaluations. Also, for cases (2-4) with number of samples greater than 1, they did better (quality of finding a good solution) than the case that samples just one point per iteration. This is because in cases 2-4, we uniformly sample points from clusters, in sequence, that are good across all three objectives (objective 1: balance, objective 2: explore, and objective 3: exploit), thereby having a good mix of points sampled uniformly across the entire design space.

2. *Different starting number of points*

Next, we varied the initial number of starting points generated using Latin Hypercube Sampling (LHS), but kept the number of infill sample points fixed to 12 per iteration. We also kept the total computational budget limited to a maximum of 300 integer data point evaluations. Figure 9 shows the convergence of the AMIEGO-MIMOS for different numbers of initial starting points. As seen from the figure on the left, different starting point sets lead to different starting "present best solution" (PBS) with a general trend (though not always) that fewer initial samples lead to worse starting PBS. Overall, all the cases converged to close-by values with the worst final solution from 151 points case (4996.3 lbs) and the best final solution from 101 points case (4968.2 lbs).



(a) Covergence plot for different initial number of data points

(b) Total evaluated data points when best solution found

Fig. 9 AMIEGO starting from different number of initial sample points

The figure (Fig. 9(b)) on the right shows the total number of discrete data points (total number of inner continuous optimization in step 2) needed to obtain the final best solution across the cases. This is the best solution after which, even though the algorithm continued to run, there is no further improvement to the present best solution. Initial sample size of 31 yields the fewest data point evaluations, while 151 starting number of sample case yields the maximum number of data point evaluations. This implies that one probably does not need a large number of initial starting samples [*].

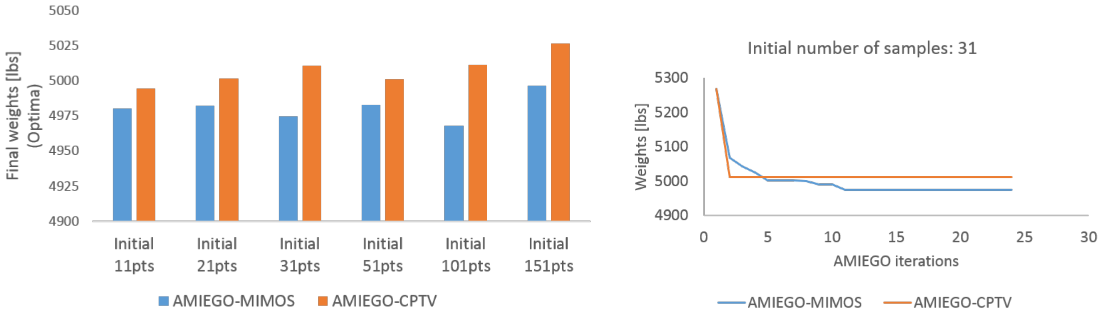
Next, we compare the performance of MIMOS and the CPTV infill criteria. Figure 10(a) shows the final solution obtained using the two different infill strategies. For the same set of initial samples and the computational budget, AMIEGO-MIMOS consistently obtains a better solution across all the 6 different cases when compared with the AMIEGO-CPTV approach. Figure 10(b) shows the convergence plot of the MIMOS and CPTV strategies for the case with 31 starting samples. After the first drop, CPTV approach could not find any further improvement to the PBS.

B. Application II: Wing topology optimization problem

1. *Problem description*

After solving the 10-bar truss problem using AMIEGO-MIMOS/CPTV, we next solve the wing topology optimization problem. A brief overview of the topology optimization problem setup is given here. The wing under consideration is the Common Research Model (CRM) developed in Ref. [37], a generic transport configuration with a wing span of 58.7 m, a mean aerodynamic chord of 7.0 m, a taper ratio of 0.25, a sweep angle of 35°, and a cruise Mach number of 0.85. The aluminum wingbox is composed of upper and lower skins, a leading and trailing spar, ribs, and T-shaped stiffeners.

[*] These runs terminated once they reached the maximum allowed data point evaluations of 300; changing this limit and allowing the algorithm to run longer could lead to a different number of total data point evaluations.



(a) Final solution obtained using MIMOS and CPTV infill strategies (b) Convergence plot for the case: initial sample size=31

Fig. 10 Comparison between AMIEGO-MIMOS and AMIEGO-CPTV

The goal here is to simultaneously optimize the layout/topology of this wingbox (in terms of rib and skin stiffener placement) and the sizing details of each component. This is graphically shown in Figure 11, where the outer surrogate-based optimizer is expected to design the internal layout through design variables which govern the number of skin stiffeners, the spatially-dependent spacing (pitch) of those stiffeners, and their orientation relative to the leading edge [†]. Similar design variables exist for the ribs. If only straight ribs and stiffeners are desired, then the number of topology design variables is 10; if curvilinear members are to be designed, the number of variables increases to 24. The current study includes only the straight ribs and stiffeners; curvilinear will be considered as part of future work.

The inner continuous optimizer (step 2) then sizes each of the wingbox components, in terms of the thickness of each shell member, the thickness of the T-stiffeners attached to each shell, and the height of those stiffeners. Several sizing "zones" are allocated down the span for each component (i.e., five total zones for the upper skins from root to tip), resulting in 75 total sizing design variables. The sizing optimization considered here is a static aeroelastic optimization, where the structural mass is minimized subject to constraints on stresses and buckling, spread over a series of trimmed aeroelastic constraints totaling to 22 problem constraints. The problem formulation for the wing topology optimization problem appears below.

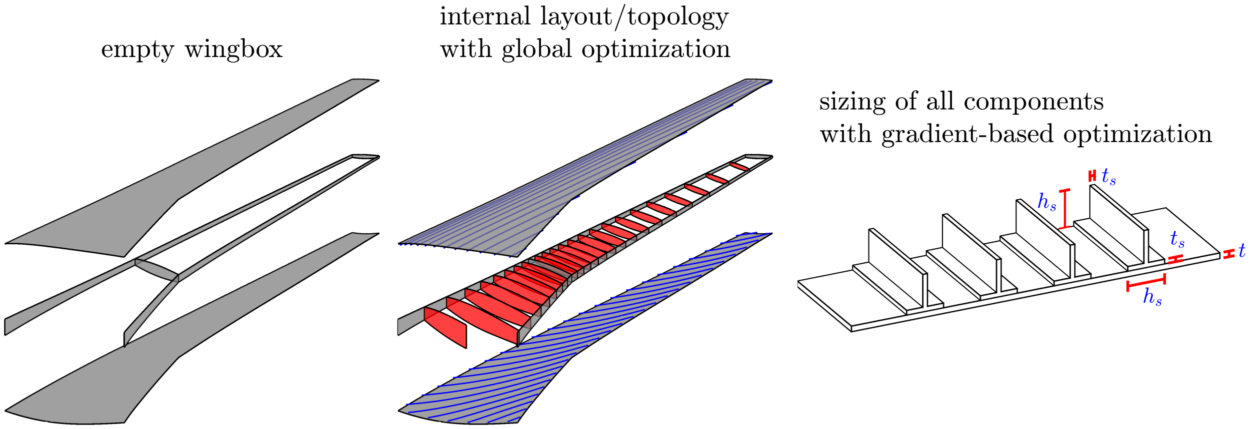


Fig. 11 An example of adding both topological detail to a transport wingbox, and structural sizing details.

A more detailed description of the problem formulation appears in Ref. [15].This type of bi-level topological/sizing wingbox design has been considered before (see for example Ref. [28]), but all past works have utilized relatively inefficient global optimizers that rely on evolutionary algorithms; the surrogate-based optimizers described above is expected to locate better designs with much fewer computational resources. Recently, Ref. [15] addressed this problem using an AMIEGO-CPTV-like approach starting with 239 initial samples, and identifies 15 infill points per iteration.

[†] Some of the variables in the outer optimization are continuous in this problem. Internally they are made discrete within AMIEGO with a resolution of 0.1

Table 1 Problem formulation for the wing topology optimization problem.

Minimize	Wing structural mass	
	(variable description)	(variable type)
With respect to	$x_{topology}$ x_{sizing}	integer continuous
Subject to	Topology variable bounds Sizing variable bounds Shell thickness constraints Stiffener aspect ratio constraints Skin and stiffener stress constraints Skin buckling constraints	

The results showed that CPTV strategy was capable of obtaining a very good solution. However, in the process, it evaluated over 1000s of integer data points. The effort here seeks to revisit the same wing topology optimization problem and investigate if a similarly comparable result can be achieved under a much tighter computational budget.

2. Result summary

In this study, AMIEGO starts with only 31 initial designs generated using Latin Hypercube Sampling (LHS) technique. Each AMIEGO iteration identifies 12 new infill points using both MIMOS and CP-only [‡] strategy and the performances are compared. We have limited ourselves to only 10 AMIEGO iterations, that would allow us to have a maximum of only 139 data point evaluations. Again, as stated earlier this is to compare the performances under a very tight computational budget.

Figure 12 shows the convergence plot for the wing topology optimization problem using both AMIEGO-MIMOS and CP-only approach. The plot reveals that CP-only has a comparable convergence rate but yields a better solution than MIMOS strategy.

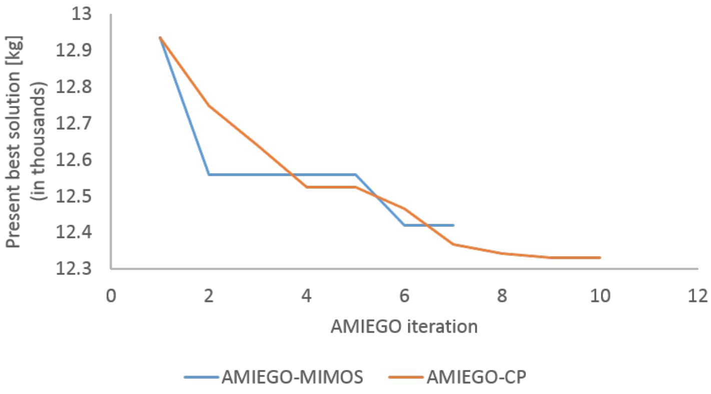


Fig. 12 AMIEGO convergence plot for the wing topology optimization problem

One plausible reasoning behind this relative under-performance of MIMOS as compared to the CP approach is that, for this wing topology optimization problem, it seems the set of very good integer design solutions are centered locally

[‡] Starting with fewer initial samples, research reveals the performance of CPTV deteriorates and the TV search hardly contributes to any improvement. From here onwards, we will only use CP search (no TV) for the topology optimization problem

in the design space thereby providing an advantage, for this specific problem, to the CP-only approach that is adept at locally exploiting around the present best solution.

To have a fair comparison between the MIMOS and the CP-only strategy, we only activate the pure exploitation case in MIMOS (only minimize objective 3: sets $g = 0$ in Eq. 3). Also, to rule out any dependencies on the initial distribution of points, we generated 10 different sets of initial sample points and obtained a mean "present best value" after each AMIEGO iteration across these 10 cases of different starting samples. Figure 13 shows the convergence plot of the average of the 10 cases after each AMIEGO iteration for both MIMOS: exploitation-only and the CP-only approach. On average, MIMOS with pure exploitation strategy shows a better convergence rate, however as the iterations progress, CP-only improves and yields a better solution compared to the MIMOS pure exploitation approach. Even after activating the pure exploitation feature in MIMOS, it could not find the solution as obtained by the CP-only approach within the first 10 AMIEGO iterations.

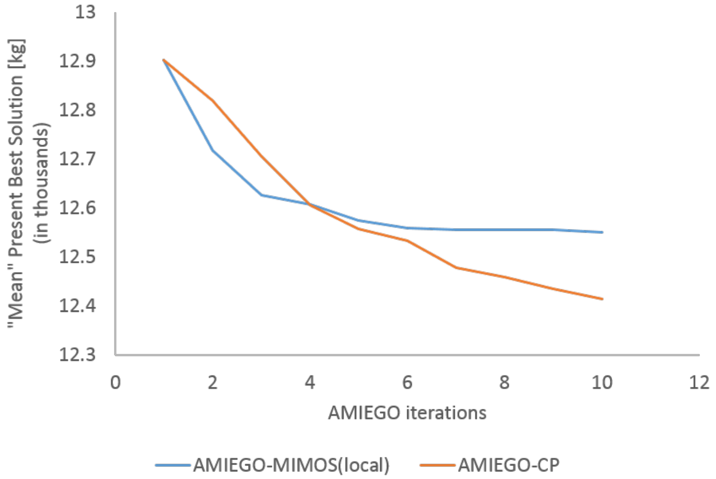


Fig. 13 AMIEGO convergence plot for the wing topology optimization problem, considering only pure exploitation in MIMOS

V. Conclusion and Future Works

The paper here extends the capability of a recently developed optimization algorithm to sample multiple points per iteration. The approach here presents a multiple infill strategy that leverages concepts of generalized expected improvement in a multi-objective formulation. The clustering technique allows uniform sampling of points giving equal importance across: exploration, exploitation and balance between exploration and exploitation. The application to engineering design problems reveal a mixed interpretation with no single strategy proving superior for all problems. For the ten-bar truss with discrete material selection choice, MIMOS consistently leads to a better solution compared to the CPTV infill criteria, while for the topology optimization problem, the CP-only approach yields a better solution, although MIMOS (pure exploitation case) shows better convergence during the initial few iterations.

Future work would include testing and comparing the MIMOS and CP approach on other complex engineering design optimization problems. The future effort will also seek to conduct research on a hybrid method that combines the advantages of these two infill criteria and informs AMIEGO, during a particular iteration, which criterion to follow.

VI. Expected Significance

The effort here seeks to deliver a methodology to solve large-scale complex engineering problems with expensive analysis tools and mixed-integer/discrete design variables, leveraging the advancement in the field of parallel computing, numerical optimization, adjoint-based methods and optimization techniques. The AMIEGO algorithm has several features that address some of the drawbacks associated with large-scale surrogate-based optimization techniques. The inclusion of parallel multiple infill strategies to the AMIEGO algorithm could lead to faster convergence to a good solution within the same wall-time as compared to the single sampling per iteration strategy.

Acknowledgments

This work is funded by NIA under the task order no 601044.

References

[1] Giles, M. B., and Niles, A. P., "An Introduction to the Adjoint Approach to Design," *Flow, Turbulence and Combustion*, Vol. 65, 2001, pp. 393–415.

[2] Martins, J. R., Alonso, J. J., and Reuther, J. J., "A Coupled-Adjoint Sensitivity Analysis Method for High-Fidelity Aero-Structural Design," *Optimization and Engineering*, Vol. 6, No. 1, 2005, pp. 33–62.

[3] Jones, 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.

[4] Gutmann, H.-M., "A Radial Basis Function Method for Global Optimization," *Journal of Global Optimization*, Vol. 19, No. 3, 2001, pp. 201–227.

[5] Regis, R. G., and Shoemaker, C. A., "A Stochastic Radial Basis Function Method for the Global Optimization of Expensive Functions," *INFORMS Journal on Computing*, Vol. 19, No. 4, 2007, pp. 497–509.

[6] Forrester, A., Sobester, A., and Keane, A., *Engineering Design via Surrogate Modelling: A Practical Guide*, Wiley Publishing, 2008.

[7] Jones, D. R., "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *Journal of Global Optimization*, Vol. 21, No. 4, 2001, pp. 345–383.

[8] Sasena, M. J., *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*, University of Michigan, Ann Arbor, MI, 2002.

[9] Roy, S., Moore, K. T., Hwang, J. T., Gray, J. S., Crossley, W. A., and Martins, J., "A Mixed Integer Efficient Global Optimization Algorithm for the Simultaneous Aircraft Allocation-Mission-Design Problem," *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech Forum*, 2017.

[10] Roy, S., and Crossley, W. A., "An EGO-like Optimization Framework for Simultaneous Aircraft Design and Airline Allocation," *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech Forum*, 2016.

[11] Roy, S., *A Mixed Integer Efficient Global Optimization Framework: Applied to the Simultaneous Aircraft Design, Airline allocation and Revenue Management Problem*, ProQuest, Ph.D. Thesis, Purdue University, 2017.

[12] Sinha, A., Malo, P., and Deb, K., "A Review on Bilevel Optimization: From Classical to Evolutionary Approaches and Applications," *ArXiv e-prints*, 2017.

[13] Bouhlel, 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.

[14] Haftka, R. T., Villanueva, D., and Chaudhuri, A., "Parallel surrogate-assisted global optimization with expensive functions – a survey," *Structural and Multidisciplinary Optimization*, Vol. 54, No. 1, 2016, pp. 3–13. doi:10.1007/s00158-016-1432-3, URL `https://doi.org/10.1007/s00158-016-1432-3`.

[15] Stanford, B., Jutte, C. V., and Coker, C., "Sizing and Layout Design of an Aeroelastic Wingbox through Nested Optimization," *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech Forum*, 2018.

[16] Booker, A. J., Dennis, J. E., Frank, P. D., Serafini, D. B., Torczon, V., and Trosset, M. W., "A rigorous framework for optimization of expensive functions by surrogates," *Structural optimization*, Vol. 17, No. 1, 1999, pp. 1–13.

[17] Basudhar, A., Dribusch, C., Lacaze, S., and Missoum, S., "Constrained efficient global optimization with support vector machines," *Structural Multidisciplinary Optimization*, Vol. 46, 2012, pp. 201–221. doi:10.1007/s00158-011-0745-5.

[18] Gengembr, 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.

[19] Tutum, C. C., Deb, K., and Baran, I., "Constrained Efficient Global Optimization for Pultrusion Process," ???? doi: 10.1080/10426914.2014.994752.

[20] Holmström, K., "An adaptive radial basis algorithm (ARBF) for expensive black-box global optimization," *Journal of Global Optimization*, Vol. 41, No. 3, 2008, pp. 447–464. doi:10.1007/s10898-007-9256-8, URL `http://dx.doi.org/10.1007/s10898-007-9256-8`.

[21] Davis, E., and Ierapetritou, M., "A kriging based method for the solution of mixed-integer nonlinear programs containing black-box functions," *Journal of Global Optimization*, Vol. 43, No. 2, 2009, pp. 191–205.

[22] Müller, J., Shoemaker, C. A., and Piché, R., "SO-MI: A surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems," *Computers and Operations Research*, Vol. 40, No. 5, 2013, pp. 1383 – 1400. doi:http://doi.org/10.1016/j.cor.2012.08.022, URL `http://www.sciencedirect.com/science/article/pii/S0305054812001967`.

[23] Müller, J., "MISO: mixed-integer surrogate optimization framework," *Optimization and Engineering*, Vol. 17, No. 1, 2016, pp. 177–203. doi:10.1007/s11081-015-9281-2, URL `http://dx.doi.org/10.1007/s11081-015-9281-2`.

[24] Hwang, 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.

[25] Hwang, 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, 2016.

[26] Roy, S., Crossley, W. A., Moore, K. T., Gray, J. S., and Martins, J. R., "Next generation aircraft design considering airline operations and economics," *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech Forum*, 2018.

[27] Wang, W., Guo, S., and Yang, W., "Simultaneous partial topology and size optimization of a wing structure using ant colony and gradient based methods," *Engineering Optimization*, Vol. 43, No. 4, 2011, pp. 433–446. doi:10.1080/0305215X.2010.493936, URL `https://doi.org/10.1080/0305215X.2010.493936`.

[28] Hansen, L., and Horst, P., "Multilevel Optimization in Aircraft Structural Design Evaluation," *Computers and Structures*, Vol. 86, 2008, pp. 104–1182.

[29] Roy, S., and Crossley, W. A., *Hybrid Approach for Multi-Objective Combinatorial Optimization in Search of Greener Aircraft*, AIAA, Virginia Beach, VA, USA, 2011.

[30] Roy, S., and Crossley, W. A., *Hybrid Multi-Objective Combinatorial Optimization Technique with Improved Compatibility between GA and Gradient-Based Local Search*, AIAA, Indianapolis, IN, USA, 2012.

[31] Nocedal, J., and Wright, S. J., *Numerical Optimization*, Springer, 1999.

[32] Gray, 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, 2010. URL `http://www.aric.or.kr/treatise/journal/content.asp?idx=134451`.

[33] Heath, 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.

[34] Bouhlel, M. A., Bartoli, N., Regis, R. G., Otsmane, A., and Morlier, J., "Efficient global optimization for high-dimensional constrained problems by using the Kriging models combined with the partial least squares method," *Engineering Optimization*, Vol. 0, No. 0, 2018, pp. 1–16. doi:10.1080/0305215X.2017.1419344, URL `https://doi.org/10.1080/0305215X.2017.1419344`.

[35] Crossley, W. A., Cook, A. M., and Fanjoy, D. W., *Using the Two-Branch Tournament Genetic Algorithm for Multiobjective Design*, American Institute of Aeronautics and Astronautics, 1999, Vol. 37, pp. 261, 267. doi:10.2514/2.699.

[36] Habib, A., Singh, H. K., and Ray, T., "A multi-objective batch infill strategy for efficient global optimization," *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 4336–4343. doi:10.1109/CEC.2016.7744341.

[37] Kenway, G., Martins, J., and Kennedy, G., "Aerostructural Optimization of the Common Research Model Configuration," *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA AVIATION Forum*, 2014.