Optimization on Metamodeling-Supported Iterative Decomposition

Kambiz Haji Hajikolaei
Product Design and Optimization Laboratory (PDOL),
School of Mechatronic Systems Engineering,
Simon Fraser University,
BC, Canada
e-mail: khajihaj@sfu.ca

George H. Cheng
Product Design and Optimization Laboratory (PDOL),
School of Mechatronic Systems Engineering,
Simon Fraser University,
BC, Canada
e-mail: ghc2@sfu.ca

G. Gary Wang[1]
Professor
Product Design and Optimization Laboratory (PDOL),
School of Mechatronic Systems Engineering,
Simon Fraser University,
BC, Canada
e-mail: gary_wang@sfu.ca

The recently developed metamodel-based decomposition strategy relies on quantifying the variable correlations of black-box functions so that high-dimensional problems are decomposed to smaller subproblems, before performing optimization. Such a two-step method may miss the global optimum due to its rigidity or requires extra expensive sample points for ensuring adequate decomposition. This work develops a strategy to iteratively decompose high-dimensional problems within the optimization process. The sample points used during the optimization are reused to build a metamodel called principal component analysis-high dimensional model representation (PCA-HDMR) for quantifying the intensities of variable correlations by sensitivity analysis. At every iteration, the predicted intensities of the correlations are updated based on all the evaluated points, and a new decomposition scheme is suggested by omitting the weak correlations. Optimization is performed on the iteratively updated subproblems from decomposition. The proposed strategy is applied for optimization of different benchmarks and engineering problems, and results are compared to direct optimization of the undecomposed problems using trust region mode pursuing sampling method (TRMPS), genetic algorithm (GA), cooperative coevolutionary algorithm with correlation-based adaptive variable partitioning (CCEA-AVP), and divide rectangles (DIRECT). The results show that except for the category of undecomposable problems with all or many strong (i.e., important) correlations, the proposed strategy effectively improves the accuracy of the optimization results. The advantages of the new strategy in comparison with the previous methods are also discussed. [DOI: 10.1115/1.4031982]

Keywords: high dimension, decomposition, iteratively, HDMR, metamodeling, sensitivity analysis

Introduction

Many optimization problems involve high-dimensional, computationally expensive, and black-box objective functions. Shan and Wang [1] coded the terminology high-dimensional, expensive, and black-box (HEB) to describe this type of problems. In the design engineering context, in general, problems with more than ten variables are considered as high dimensional, if function evaluations are computationally expensive [2]. Expensive problems typically have objective functions in the form of software simulations, such as finite-element analysis (FEA) and computational fluid dynamics (CFD). They are expensive because each function evaluation takes at least 1 order of magnitude more time than running one iteration of the optimization algorithm. Using software simulation makes the objective function black-box, which means its functional form, (non)linearity with respect to each variable, and variable correlations are unknown. Though finite-difference methods can be applied to get gradient information, it is found that such gradients are often noisy, erroneous, and expensive when computed from FEA or CFD processes [3]. Moreover, when the dimensionality of the problem grows, calculating the derivatives needs numerous function calls, which is not feasible considering the computational cost.

Beside finite difference, there are other methods, such as complex-step [4] and adjoint methods [5], used mostly in CFD optimization. However, they are not suitable for strict black-box simulation-based optimization because of different reasons. The complex-step method first assumes that the function is complex analytical (holomorphic), and second, needs function values of complex variables which are not available in our case. On the other hand, the adjoint methods require partial derivatives [6], which are usually complex, and most CAE software tools do not provide such information to the users. In this paper, it is assumed that the objective function is a black-box function and the gradient of the function is not available to be used in optimization. All the above challenges make it difficult to solve HEB problems [7–9].

Various strategies are used for addressing high dimensionality in science and engineering fields [1]. Beside parallel computing and increasing computational power, which are the clear choices in all conditions, reducing design space, screening significant variables, visualization, and decomposing design problem into subproblems are the most commonly used strategies. Screening methods, such as sensitivity analysis, analysis of variance (ANOVA), and PCA, are used to judge the importance of input variables in design problems. Then, the less important input variables are removed and the problem is simplified by reducing the dimensionality [10–12]. Design space reduction, in general, refers to any method that focuses on a subregion of the design space. Shrinking the ranges of design variables leads to shrinkage of the design space and results in more efficient optimization. A common space reduction approach is sampling and evaluating a limited number of sample points then reducing the design space based on feedback information from modeling on these sample points. Wang et al. [13] proposed the adaptive response surface method and the fuzzy clustering based approach [14], in which the design space was iteratively reduced. Visualizations are used to visually represent the design problems to the user and can also be used for design space or dimensionality reduction through interaction with the user. Winer and Bloebaum [15,16] proposed a method

[1]Corresponding author.

Contributed by the Design Automation Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received December 2, 2014; final manuscript received October 19, 2015; published online December 8, 2015. Assoc. Editor: Christopher Mattson.

that allows a designer to make decisions before, during, or after an analysis or optimization via a visual environment to steer the solution process. Although all the mentioned methods are useful in different situations, decomposition is identified as one of the most promising approaches for tackling high-dimensional problems.

Decomposition recommends subproblems of smaller dimension that can be optimized separately, instead of optimizing all the dimensions together. After optimization of subproblems, all subcomponents of the main input vector are recombined and the system is rebuilt through combination of obtained solutions [17]. In this way, the required effort for optimization is substantially reduced. The fact that the subproblems generated by decomposition can be solved in parallel has made the partitioning and coordination strategies of interest for decades. The challenging issue is how to decompose a problem, as opposed to how to solve already partitioned problems. Analytical target cascading is a hierarchical, decomposition-based strategy for large-scale optimization [18], which optimizes the objective of each subproblem while minimizing deviations between design targets. Michelena et al. [19] decomposed the problem into different partitions of variables and constraints using hierarchical overlapping coordination. Alison et al. [20] implemented partitioning and coordination strategies simultaneously by formulating them as an optimization problem. Alexander et al. [21] used radial basis function to reduce the dimensionality of vector-valued coupling variables.

Cooperative coevolution (CC) methods have also been proposed to leverage the capability of evolutionary algorithms (EA) in dealing with high-dimensional problems by decomposing them into lower order problems. Traditionally, decomposition strategies in CC were *one-dimensional based* and *splitting-in-half* [22–24]. Yang et al. [25] used random grouping scheme plus an adaptive weighting system for decomposition in CC. The work has been elaborated in Ref. [26], adopting particle swarm optimization position rules using Cauchy and Gaussian distributions for sampling new sample points and adaptively determining the sizes of the random grouping. Different than the standard iterative CC methods, Omidvar et al. [27] used the fitness contribution of various subproblems in splitting the computational budget between the subproblems. In this way, more computational resources are allocated to the subproblems with higher amount of contribution to the fitness function. Multilevel cooperative coevolution [28] and multilevel Softmax [29] adaptive methods are used for fully separable continuous optimization problems. Differential grouping method is used for uncovering the underlying interaction structure of the input variables in Ref. [30]. However, the number of function calls before reaching an acceptable grouping scheme is still too high for being used in computationally expensive problems. Also, because of not considering the intensities of the correlations, the methods are not capable of optimizing nonseparable problems. Singh and Ray [31] introduced CCEA-AVP that decomposes nonseparable optimization problems based on the relations between the design variables. CCEA-AVP is one of the methods used in Sec. 4 to test the performance of the current work. Mahdavi et al. [32] used metamodeling in CC decomposition but again the intensities of the correlations are not calculated. Similar approach is used in Ref. [33] without using EA. Radial basis function (RBF)-HDMR metamodeling is used to find the internal structure of a black-box function and if the problem variables are decomposable, subproblems are recommended for optimization.

In Ref. [34], after building the metamodel, sensitivity analysis has been performed on the metamodel to quantify the correlation intensities. Then, weak correlations are omitted and a new decomposition scheme is generated with only strong correlations being used for decomposition. It is worth mentioning that *decomposition* and *optimization* steps have been performed separately in Ref. [34]. If the problem has undecomposable structure with all strong correlations, the decomposition phase suggests one large group including all the variables for optimization. In this case, the sample points used for decomposition are basically wasted. Also, the rigidity of the two-step strategy in building only one decomposition scheme increases the possibility of missing the global optima, if the decomposition is not correct. On the other hand, to ensure adequate accuracy of the decomposition, extra expensive sample points are needed. Moreover, the metamodel used in Ref. [34], RBF-HDMR, needs structured sampling of the problem, which is often not achievable in engineering problems.

In this paper, decomposition is not separated from optimization; the sample points used in the optimization phase are reused for decomposition. In this way, even if the problem is undecomposable, all the sampling costs are used for finding better results. Moreover, a newly developed metamodel, PCA-HDMR, is used for metamodeling and sensitivity analysis which does not demand structured sampling and can be built with nonuniform samples. The algorithm is named optimization on metamodeling-supported iterative decomposition (OMID). The rest of this paper is organized as follows. First, the metamodel HDMR and the metamodel-based decomposition are briefly described. Then, the proposed decomposition strategy and the optimization approach are explained. The Method Testing section provides results of the proposed method on different benchmark functions, along with an engineering design problem. Finally, conclusions are given.

Background

High Dimensional Model Representation (HDMR). HDMR is an approximation method, first introduced by Sobol [35] and then elaborated by Rabitz and Alis [36], for representing high-dimensional black-box functions. It models a high-dimensional problem as a superposition of lower order functions. It consists of terms for the individual and joint contribution of the input variables to the system output. ANOVA-HDMR [36] and cut-HDMR [37] are two main types of HDMR, which have their own disadvantages in modeling black-box functions. To overcome these disadvantages, random sampling-HDMR (RS-HDMR) [38] and PCA-HDMR [39] are introduced as a modified version of ANOVA-HDMR that needs only one set of randomly scattered sample points. They both use linear combinations of basis functions to build the HDMR components with a general form of

$$f(x) = c_0 + \sum_{i=1}^{d}\sum_{k=1}^{s} c_i^k \phi_i^k(x_i) + \sum_{i<j}^{d}\sum_{k=1}^{s'} c_{ij}^k \phi_{ij}^k(x_i,x_j) + \cdots \quad (1)$$

where $\{\phi_i^k(x_i)\}_{k=1}^{s}$ is a family of linearly independent basis functions for univariable functions of x_i on $[0\ \ 1]$ and $\{\phi_{ij}^k(x_i,x_j)\}_{k=1}^{s'}$ is similarly defined as a family of linearly independent basis functions for bivariate functions of x_i and x_j on $[0\ \ 1]^2$ and so on. The problem of finding HDMR components will change to finding the unknown coefficients of the basis functions $(c_0, c_i^k, c_{ij}^k, \ldots, c_{i_1 i_2 \ldots i_L}^k)$. To find PCA-HDMR unknown coefficients, a new set of coordinates are defined as the linear transformation between the original coordinates with minimum possible amount of variation using the principal component analysis (PCA). A limitation of using PCA-HDMR is that a minimum number of sample points are required to build a PCA-HDMR. To overcome the problem of required minimum number of sample points, it is recommended to just use the important variable correlations. When only a fraction of correlations are considered, the model is called partial PCA-HDMR. The proposed OMID algorithm uses partial PCA-HDMR. More details on PCA-HDMR metamodeling algorithm and formulation can be found in Ref. [39].

Structural and Component Correlation Matrices. The structural matrix (SM) and component correlation matrix (CCM) represent the correlations among the variables of a function in HDMR representation. SM is a $d \times n$ matrix showing all the existing terms of HDMR, in which n is the number of HDMR terms and d is the dimensionality. An example is shown in the below equation

$$SM = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1
\end{bmatrix} \tag{2}$$

Each row corresponds to a variable and each column corresponds to one of the component terms in the HDMR. Each element in the SM takes "0" or "1"; in each column, 0 means that the corresponding variable does not participate in the component term and 1 means that the variable does participate. For example, the first column $[0_1, 0_2, \ldots, 0_d]^T$ denotes the constant component term, in which no variable is participating; $[0_1, 0_2, \ldots, 1_i, \ldots, 0_d]^T$ represents the first-order component term $f_i(x_i)$; $[0_1, 0_2, \ldots, 1_i, \ldots, 1_j, \ldots, 0_d]^T$ indicates the existence of the second-order component term $f_{ij}(x_i, x_j)$, and so on. CCM is a square matrix defined as

$$CCM_{d\times d} = [m_{ij}], \quad i = 1, 2, \ldots, d; j = 1, 2, \ldots, d \tag{3}$$

where $m_{ij} = 1$ if a correlation exists between variables x_i and x_j in an HDMR formula for a particular problem; otherwise $m_{ij} = 0$. The zero and one elements scatter in the CCM shows the structure of the underlying black-box function. CCM matrix can be made from SM matrix but the reverse is not possible. Equation (4) shows the CCM, corresponding to SM matrix in Eq. (2). It can be seen that the variables $\{x_1, x_2, x_3, x_5, x_7, x_8, x_9, x_{10}\}$ are correlated, and variables x_4 and x_6 are independent. The same grouping strategy is used in this paper

$$\text{CCM} = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1
\end{bmatrix} \tag{4}$$

Metamodel-Based Decomposition. Recently, the authors' team proposed an algorithm for decomposing high-dimensional problems to smaller subproblems using a combination of metamodeling and sensitivity analysis [34]. The metamodel-based decomposition algorithm [34] proposes to decompose the original high-dimensional problem first and solve the decomposed subproblems, instead of the original undecomposed problem. The method builds a metamodel (RBF-HDMR) of black-box function, then uses sensitivity analysis for quantifying the intensities of the correlations among the variables. If the sensitivity index of a correlation is less than a predefined criterion, the correlation is omitted from the SM of the problem. The modified SM and CCM are used for decomposing the problem. For example, if the modified CCM matrix is found similar to the one in Eq. (4), instead of optimizing the ten-dimensional original problem, the three subproblems of $(x_1, x_2, x_3, x_5, x_7, x_8, x_9, x_{10})$, (x_4), and (x_6) are optimized. More details about the procedure can be found in Ref. [34]. This work is an improvement of the above-mentioned metamodel-based decomposition strategy by addressing its disadvantages, which are listed below:

(1) RBF-HDMR metamodeling needs structured samples, which are not always available for engineering problems.
(2) To find the structure of the objective function and the sensitivity of the correlations, a certain number of sample points are needed, which is in addition to the optimization costs.
(3) When the problem is complicated, to find all the correlations, RBF-HDMR needs an excessive number of sample points that may go even beyond the allowed sampling budget.
(4) In cases when all correlations are strong and the problem is deemed undecomposable, the metamodeling basically wastes the valuable sample points without helping the optimization.
(5) In cases when the decomposition is not accurate, optimization may miss the global optima and there is no chance to improve the decomposition scheme.

Proposed Method

In this section, steps of OMID algorithm are presented. The main idea of the proposed algorithm is decomposing the problem to smaller subproblems iteratively during the optimization process, based on the knowledge of the problem obtained until the current iteration. In this paper, PCA-HDMR's capability of working well with nonuniform samples is employed. RBF-HDMR sampling is omitted and instead the samples used during the optimization process are reused to build a PCA-HDMR model for the purpose of sensitivity analysis. Figure 1 shows the flowchart of OMID. Beside the initial sampling unit, the algorithm has four main units which are repeated during the optimization process and are shown in the figure: correlation identification (CI), metamodeling prediction (MP), subproblem optimization (SO), and global search (GS).

The procedure starts with generating initial sample points (X_{Init}). The samples are randomly scattered in the search space and used to build a PCA-HDMR metamodel. There is a tradeoff between the efficiency and accuracy when choosing the number of initial sample points (N_{Init}). If N_{Init} increases, the initial estimation of the variable correlations and their intensities will be more accurate but the total cost of the algorithm will increase as well. On the other hand, if a small number is chosen, a few first iterations are not going to properly predict the correlations and their intensities. Basically, the first few iterations waste the expensive sample point in searching wrong subproblems because of less accurate approximated intensities. The other reason can be thought in biasing the samples when reusing optimization sample points. A reasonable number of initial randomly scattered sample

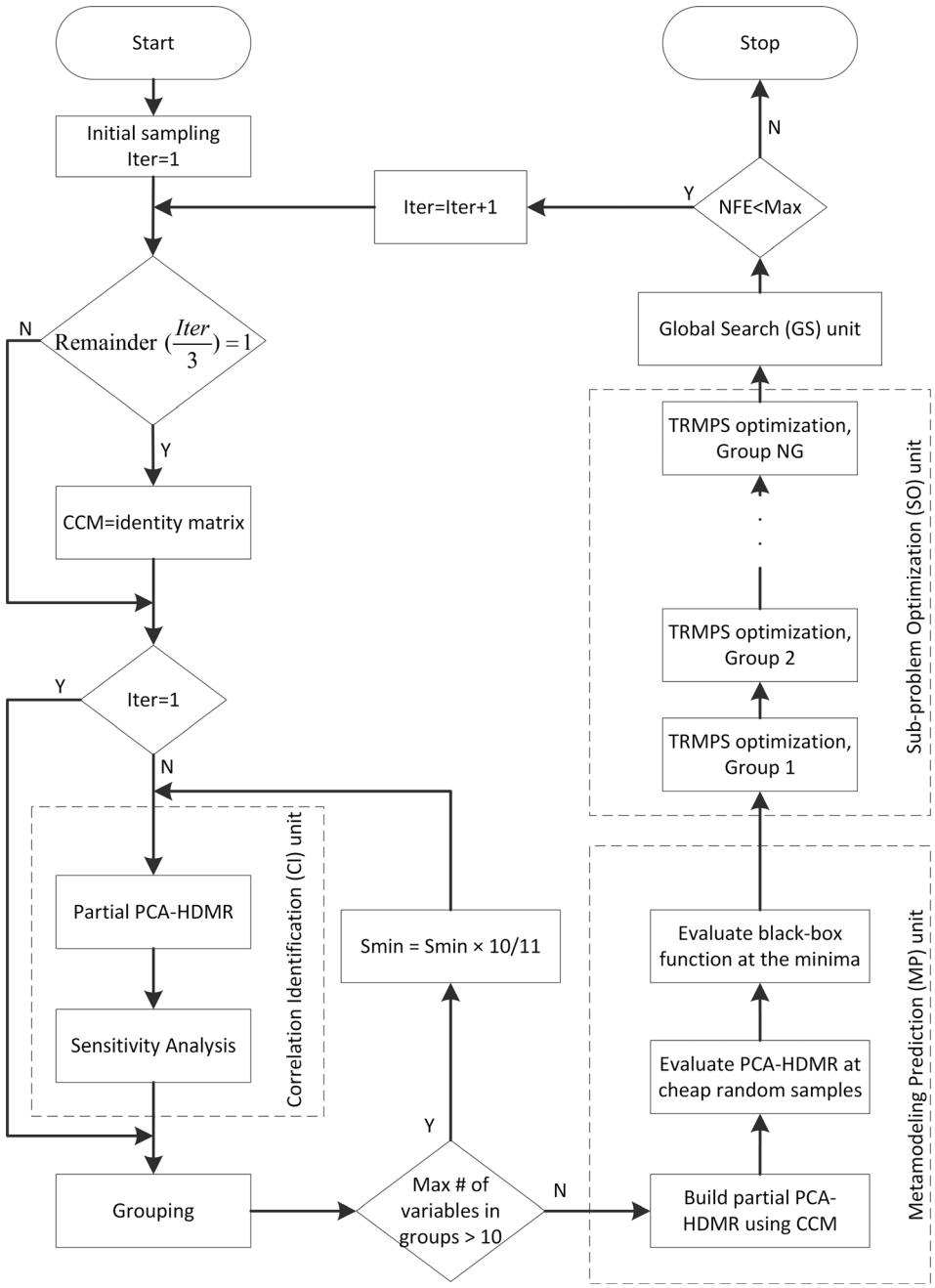

Fig. 1 OMID flowchart

points will help to explore the search space as much as possible. Different values are tested for different dimensionalities and finally the number of initial samples is set to be a linear function of the number of variables ($N_{\text{Init}} = 10 \times d$). The value can be changed by the user based on the needed accuracy and efficiency.

As shown in the figure, except at the first iteration, in which the CCM matrix is set to be an identity matrix, the rest are using the last updated CCM from the previous iteration. All the sample points evaluated by the black-box function until the current iteration are given to the CI unit to update the CCM matrix. Pseudo-code below summarizes how the CI unit works. The output is the updated CCM matrix

(1) Select the number of correlations to be added n_{corr}.
(2) Set zero elements of CCM equal to one, one-by-one.
(3) Build the corresponding partial PCA-HDMR metamodels.
(4) Record the R-squared values.
(5) Select the n_{corr} elements with the maximum R-squared values.
(6) Build partial PCA-HDMR with CCM including all n_{corr} elements.
(7) Perform sensitivity analysis on the model.
(8) Neglect the correlations with a sensitivity indices less than S_{min}
(9) Report the updated CCM

The first step of the CI unit is selecting the number of correlations to be added to CCM. For adding one correlation, $n_{\text{comp}} \times n_{\text{basis}}$ sample points are needed, where n_{comp} and n_{basis} are the order of components and number of one-variable basis functions used to model the component, respectively. In this paper, both

values are equal to 2. It means that PCA-HDMR goes to second-order correlations and two basis functions (first- and second-order polynomials) are used. However, sometimes adding too many correlations leads to having all-correlated variables (i.e., one group of variables) too soon, which defeats the purpose of decomposition. Therefore, two caps are used to avoid this situation. The first cap is for the number of added correlations (n_{corr}). Theoretically $(N_{New\,Samples}/(n_{comp} \times n_{basis}))$ correlations can be added to CCM at every iteration but the number n_{corr} is capped to be the number of variables, d. In this way, the CCM is not likely to lead to the all-correlated stage too soon. In cases that there is prior knowledge of the structure/complexity of the problem, n_{corr} can be tuned accordingly. The second filtering step happens in sensitivity analysis that is explained later in this section.

The next step in the CI unit is setting, one-by-one, the zero elements of the CCM from the previous iteration equal to one, then building the corresponding PCA-HDMR metamodel for each newly added variable correlation. After building all partial PCA-HDMR models, the R-squared values of the metamodels are recorded. Assuming that there are n_z zero elements in the CCM matrix, n_z partial PCA-HDMR metamodels are built and n_z R-squared values are recorded. The closer the value of R-squared to one, the more accurate is the approximated model. Therefore, the metamodels with higher values of R-squared are more likely to be the ones with most important correlations. n_{corr} of the tested CCM elements with the greater R-squared values are selected as the candidates of new correlations and are added to the previous iteration's CCM.

The last step of CI unit is the sensitivity analysis of the selected correlations. As mentioned before, to avoid reaching all-correlated stage prematurely, it is tried to pick the best correlations and complete the CCM gradually. Moreover, in cases with mostly separable variables, it is not desirable to force new correlations to the problem structure. Therefore, at this stage, the sensitivity analysis is performed on the added correlations in order to filter unimportant ones. This is the second filtering step mentioned previously. A partial PCA-HDMR is built using all the sample points with a CCM consisting of previous iteration's correlations plus the selected correlations from the last step. Sensitivity analysis is performed and the correlations with small sensitivity indices, known as weak correlations, are deleted from CCM. Similar to Ref. [34], a parameter called S_{min} is defined which can be constant or varying during the algorithm. Different from Ref. [34], S_{min} is not determined by the user manually, because the decomposition is performed several times during the algorithm and decomposition and optimization phases are intertwined in OMID. S_{min} is defined in this work as follows:

$$S_{min} = \frac{0.1}{d \times \text{Iter}} \tag{5}$$

The value is inversely related to the number of variables because the number of correlations increases with the problem dimensionality and the summation of all sensitivity indices is equal to one. S_{min} is also set inversely proportional to the iteration number of the process. The process starts with a large S_{min} value, which means it only accepts very strong correlations. Therefore, during the first few iterations, subproblems have smaller size (i.e., less number of variables in a group). Progressively, S_{min} decreases and the algorithm accepts weaker correlations and leads to subproblems with more input variables. At this stage, a limit can be considered for the maximum number of variables that a subproblem can have, depending on the method used for SO. In this paper, the maximum number of variables in subproblems is set to be ten. It is assumed that with the specified number of sample points for SO s $(20 \times d)$, the optimization algorithm works well for problems of ten or less number of variables. This limit can change, by changing the optimization algorithm or changing the number of allowed sample points.

An updated CCM is the output of the CI unit and is used for grouping the variables. Note that at the first iteration ($i = 1$), all variables of the problem are assumed separable (i.e., there is no strong correlation among variables). Therefore, the CCM matrix is set equal to the identity matrix of size d, the problem dimensionality, and the CI unit is bypassed. Therefore, the problem is broken to d one-variable subproblems. The CCM matrix resets to the identity matrix every three iterations. It has been shown in the flowchart by Remainder $(\text{Iter}/3) = 1$. In other words, if the remainder of the iteration number divided by three is equal to one, the CCM matrix becomes identity matrix. First, this refreshes the CCM matrices. Reaching subproblems with a large number of input variables is against the decomposition. Second, this is to rejudge the previously selected correlations with new information and delete those incorrectly chosen. Usually during the first few iterations, the number of sample points is not enough to correctly judge the correlation intensities. Refreshing the CCM matrix improves the correlation selection accuracy.

Before proceeding to the SO unit, a partial PCA-HDMR metamodel is built based on the CCM matrix. At the first iteration, because of not considering any correlation between the variables, a first-order PCA-HDMR is built. Then, a large number of random sample points are generated and the metamodel is evaluated at those points. The point with the minimum metamodel value is evaluated by the black-box objective as a prediction for the actual optimum point.

The next step is optimization of the subproblems. At this stage, any algorithm can be used for optimization of the subproblems (TRMPS is used in this paper). Except for the first iteration at which the variables are considered as separable, at the rest of the iterations, the subproblems may have more than one variable. In these cases, the subproblems are optimized in the SO unit based on their number of variables. The optimization starts with the subproblem with the largest number of variables and continues until all the subproblems are optimized. In each case, the sampling budget is a linear function of the number of variables in the subproblems $(20 \times d)$. When optimizing the first subproblem, the variables of the rest of the subproblems are set to be constant values. The constants are chosen to be at the best location so far. For the first iteration ($i = 1$), the best sample point during the initial sampling and MP unit is chosen for this purpose. For other iterations, the best optimum locations found during previous iterations are selected.

After optimizing the subproblems, contour-based discriminative sampling [40] is performed (GS unit) for two main reasons:

(1) *Global Search*: All the subproblem optimizations using TRMPS are performed in subspaces of the entire region, which act as the exploitation phase. Contour-based discriminative sampling searches in the entire search space with the goal of finding new desirable search regions (exploration).
(2) *Unbiasing the metamodeling data*: Although PCA-HDMR works using nonuniform sampling, concentration of all the samples in small subregions can affect the accuracy of the metamodel in other regions. Therefore, contour-based discriminative sampling which is performed in the entire search space will help maintain the global accuracy of PCA-HDMR.

Again, there is a tradeoff between accuracy and efficiency in choosing the number of points used in GS (N_{GS}). Increasing this number will increase the probability of finding favorable search regions (i.e., better exploration) but will also increase the total cost of optimization. In this paper, N_{GS} is set to be a linear function of the dimensionality $(N_{GS} = 5 \times d)$. However, if priori knowledge of the problem is available, the number could be changed accordingly. For example, if it is known that the objective function is multimodal with many local

Table 1 CCEA-AVP parameter settings

Parameter	Value
Population size	50
Number of generations	20
Crossover probability	1.0
Mutation probability	0.1
Crossover index	15
Mutation index	20
Maximum partitions for CCEA-AVP	10
Correlation threshold, T	0.6

minima, N_{GS} should be increased to empower the exploration phase, with the goal of not getting trapped in local minimum regions.

Different stopping criteria can be chosen according to the application in which the algorithm is used. In this paper, the number of sample points is used as the stopping criterion that makes the algorithm comparison with other methods easier. It is worth mentioning that after every TRMPS run on subproblems, the total number of used sample points is compared with the criterion. If the number is more than or equal to ($NFE_{max} - N_{GS}$), the rest of the subproblems are not optimized and the algorithm jumps to GS unit.

Method Testing

In this section, different types of benchmark functions and a real-world engineering problem are chosen to test the OMID algorithm. The test problems [41,42], as defined in the Appendix, are chosen with different functional shapes, number of variables, and intensities of variable correlations. Four different categories of problems are identified according to intensities of variable correlations [34]

(1) *Decomposable problems* (*type 1*): In these problems, variables are correlated in subproblems that are of small scale. In other words, without ignoring any correlation, the problems are decomposable to small subproblems. Test problem 1 (in Table 2) is of this type.
(2) *Undecomposable with weak correlations* (*type 2*): In these problems, variables are correlated and undecomposable. However, the variable correlations are too weak. In other words, the correlation terms are not as important as independent terms and good results can be obtained by just optimizing the variables separately. Test problem 2 (in Table 2) is of this type.
(3) *Undecomposable with a mix of weak and strong correlations* (type 3): In these problems, the variables are all correlated but the intensities of the correlations are different. OMID algorithm (CI unit) identifies the intensities and ignores the weak correlations based on the S_{min} criterion and the decomposed subproblems are optimized. Test problems 3–14 (in Table 2) are of this type.
(4) *Undecomposable with strong correlations* (*type 4*): This type of problem has correlations with similar strong significance. Test problems 15–17 (in Table 2) are of this type.

Table 2 shows the results of OMID algorithm compared with TRMPS, GA, CCEA-AVP, and DIRECT. The tests are repeated ten times for each benchmark function, and the averages of the obtained optimum values as well as the standard deviation are presented. The benchmark functions are shown in the Appendix. Because DIRECT is deterministic and all runs give same results, there is no standard deviation to report. Note that the maximum number of function evaluations is set to be 5000 for TRMPS, GA,

Table 2 OMID algorithm results compared with TRMPS, GA, CCEA-AVP, and DIRECT

			GA		DIRECT		TRMPS			CCEA-AVP			OMID	
#	Problem	Optimum fval	Mean fval	STD of fval	nfe	fval	Mean nfe	Mean fval	STD	nfe	Mean fval	STD	Mean fval	STD
1	20D SUR-T1-16	0	1.169	0.6658922	5381	152.4	5000	0.745	0.362	5260	35.8	22.6531	0.201	0.1075
2	F16	25.875	45.3	4.182	5355	26.37	700	25.9276	0.1485	738	29.7	1.8852	25.8852	0.0064
3	10D PUR-T1-13	0	1.418	0.3853	5151	1.36	4153	9.84	26.2	5200	0.0895	0.1460	0.0000	0.0000
4	20D PUR-T1-13	0	31.35	24.91	5245	5.28	5000	8.399×10^5	1.45×10^6	5200	5.2×10^6	1.47×10^7	0.805	1.6064
5	30D PUR-T1-13	0	243.6	130.4	5149	252	5000	3.189×10^8	3.55×10^8	7100	4.14×10^8	7.07×10^8	657	1184.4169
6	10D SUR-T1-14	0	9.979	23.447	23903	9.75×10^9	5000	1.086	0.201	5200	2.15	1.8323	1.2207	0.2734
7	20D SUR-T1-14	0	7.97×10^5	9.71×10^5	8919	8.58×10^{13}	5000	2.192	0.848	5100	85.1	55.7391	2.41	1.4306
8	30D SUR-T1-14	0	7.510×10^8	7.152×10^8	8071	1.01×10^{16}	5000	3.536	1.536	7100	293	199.2746	54.8	106.0589
9	10D SUR-T1-14 modified	0	3.0580	2.1545	5101	1.4593	5000	1.14	0.3195	5200	6.26	4.4934	1.18	0.0779
10	20D SUR-T1-14 modified	0	54.8852	29.4094	5017	5.9331	5000	11.7	9.4373	5100	118	87.8771	9.27	9.8144
11	30D SUR-T1-14 modified	0	499.5408	225.3103	5089	171.8373	5000	56.2	16.18926	7100	1560	973.9042	22.8	31.3537
12	10D Zakharov	0	1.83	1.73	5139	34.99	3532	0.371	0.764	5170	21.2	11.3531	0.203	0.1661
13	20D Zakharov	0	28.6	122.88505	5175	109	5000	0.235	0.1888	5090	125	36.0097	123	66.4936
14	30D Zakharov	0	8.067	2.776	5317	179.6	5000	31.03	14.71	6860	241	68.2861	284	72.0840
15	10D Rosenbrock	0	6.30	1.3931	5139	7.85	3828.33	5.548	1.99	5200	55.9	41.1659	19.7823	29.4380
16	20D Rosenbrock	0	44.73	19.915	5331	17.89	5000	17.98	1.41	5100	140	41.2827	41.2	29.7582
17	30D Rosenbrock	0	99.83	24.21	5605	27.9	5000	21.53	7.54	7090	165	37.8739	117	59.4922

and DIRECT. However, TRMPS converged before reaching 5000 function evaluations in some cases. For OMID and CCEA-AVP, the maximum number of function evaluations is set to be the value that TRMPS used. In this way, the comparison is fair between TRMPS, CCEA-AVP, and OMID and also all of them are using less number of sample points than GA and DIRECT. The used sample points in all tests are 5000 in GA and it is thus not reported in the table. In some cases, it is seen in the table that DIRECT and CCEA-AVP go beyond the specified number of sample points and it is because we let the algorithms complete the last iteration. The main comparison is between TRMPS and OMID because the subproblems of OMID are optimized by TRMPS. The parameters used for TRMPS are identical to that used in Ref. [43]. It should be mentioned that the method is named TRMPS2 in the original paper [43] and it is called TRMPS in this paper for convenience.

GA and DIRECT are chosen to compare OMID with these well-known methods in the literature and also CCEA-AVP is chosen for this purpose because it is a decomposition-based optimization method. There are several types of GA in the literature. In this paper, the GA available in the MATLAB optimization toolbox is used and the parameter settings are the same as those used in Ref. [43] with the crossover rate, mutation rate, population size, and stall generation limit being 0.6, 0.05, 100, and 49, respectively.

The parameters chosen for CCEA-AVP are shown in Table 1. The only exception is the setting for test problem 2 in which the population size and number of generations are set to be 10, in order to make sure that the algorithm goes through a couple of iterations to reach approximately 700 function calls to be comparable with TRMPS. If the setting in Table 1 is chosen, then the algorithm would not go beyond the first iteration.

The result of problem 1 (SUR-T1-16) shows that OMID works well for type 1 problems. The average function value is reduced from 0.745 by TRMPS to 0.201 by OMID. Both OMID and TRMPS are remarkably better than GA, CCEA-AVP, and DIRECT for this problem. For better understanding of the procedure, the subproblems' variables considered in OMID in each iteration are studied, for one of the independent runs. By looking at the problem formulation, it is clear that the variables are correlated in groups of four and the problem consists of five-independent subproblems (x_1, x_6, x_{11}, x_{16}), (x_2, x_7, x_{12}, x_{17}), (x_3, x_8, x_{13}, x_{18}), (x_4, x_9, x_{14}, x_{19}), and (x_5, x_{10}, x_{15}, x_{20}). Figure 2 shows that the decomposed subproblems were assumed and optimized by OMID during different iterations of a single run for SUR-T1-16. In this figure, black and white pattern fills of the variables' rectangles represent different groups of variables with the corresponding variable indices to their left.

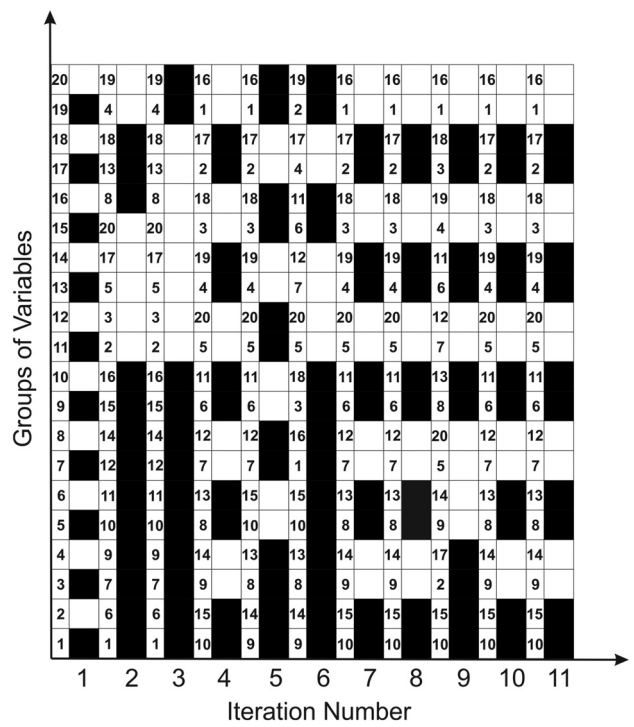

Fig. 2 Decomposed subproblems during OMID procedure for problem #1

The first iteration starts with optimizing the variables independently. After the first iteration is completed, the optimization points are reused for identifying the important correlations. At iteration 2, the subproblems assumed by OMID are (x_1, x_6, x_7, x_9, x_{10}, x_{11}, x_{12}, x_{14}, x_{15}, x_{16}), (x_2, x_3, x_5, x_{17}, x_{20}), (x_8, x_{13}, x_{18}), and (x_4, x_{19}), which are different from the actual subproblems. At iteration 3, subproblem #2 is also connected to subproblem #3 and the algorithm optimized three subproblems. Iteration 4 is the one that the CCM matrix is reset and the new subproblems are getting more accurate. As mentioned in the Proposed Method section, the CCM matrix is reset every three iterations and new correlations are made, based on the existing sample points so far. It is the reason why the numbers of variables in groups are increasing when the iteration grows until third iteration but at iteration 4, the numbers of variable in groups are less than that at iteration 3.

At iteration 4, predicted ten different subproblems are (x_{10}, x_{15}), (x_9, x_{14}), (x_8, x_{13}), (x_7, x_{12}), (x_6, x_{11}), (x_5, x_{20}), (x_4, x_{19}), (x_3, x_{18}), (x_2, x_{17}), and (x_1, x_{16}), which are shown with black and white pattern fills in Fig. 2. At the first look, it seems that CI missed many correlations and instead of five subproblems, assumed ten smaller subproblems. But by a closer look at the problem formulation, it can be seen that there are two summations with second-order terms and two summations with fourth-order terms. The correlations found by OMID are exactly the fourth-order correlations. In other words, OMID found the most important terms and neglected the less important (second order) terms, based on the formulation defined by Eq. (5) for $S_{\min}$ criterion. It is clear that if a greater value of $S_{\min}$ is considered as criterion, more correlations would be retained and subproblems with more variables would be identified. As shown in Table 2, the chosen criterion is working properly and OMID obtained smaller average optimum value than TRMPS with a smaller standard deviation, and they are both better than GA, CCEA-AVP, and DIRECT results. The number of function evaluations in DIRECT is more than 5000, because we allow DIRECT to complete the last iteration, but still reaches a poorer optimum than OMID. As shown in the figure, although $S_{\min}$ is decreased from iteration 4 to iteration 5, iteration 5 keeps almost the same subproblems and just combines two groups together.

Iteration 6 connects a few groups together and predicts the subproblems (x_9, x_{14}, x_8, x_{13}, x_{10}, x_{15}, x_1, x_{16}, x_3, x_{18}), (x_5, x_{20}, x_7, x_{12}), (x_6, x_{11}), (x_4, x_{17}), and (x_2, x_{19}). Iteration 7 is again the reset iteration (third iteration after iteration 4), which returns to the same pattern as iteration 4 and iteration 8 keeps the same pattern. Note that based on Eq. (5), the $S_{\min}$ criterion is proportional to the reverse of the number of iterations. Iteration 9 again combines a few groups and then iteration 10 resets it to the same pattern as iteration 8, which is kept in the last iteration as well.

Problem 2 is of type 2: The correlations are not identified by RBF-HDMR metamodeling in Ref. [34]. The reason why the number of sample points is not 5000 in this problem is that TRMPS converges around 700 sample points with good results. We let GA and DIRECT continue until 5000 sample points but still they cannot reach favorable results in comparison with TRMPS and OMID. For being fair in comparing the results, OMID and CCEA-AVP are constrained to 700 sample points

similar to TRMPS and OMID converged with better and more consistent results than TRMPS.

Problems 3–14 are of type 3: By looking at the function PUR-T1-13 (problems 3–5), the importance of the correlations among the variables is increasing when their index number (i) is increasing. For example, the correlation between variables (x_9, x_{10}) is much more important (i.e., higher sensitivity index) than the correlation between variables (x_1, x_2). It can be seen by comparing the coefficients of the terms having variables (x_9, x_{10}) with the coefficients of the terms having (x_1, x_2). As shown in Table 2, OMID yields much better results than TRMPS, GA, CCEA-AVP, and DIRECT in all 10-, 20-, and 30-dimensional cases for this problem.

By looking at the function SUR-T1-14 (problems 6–8) formulation, it is clear that the variables are all correlated with variables beside them like a chain (i.e., variable x_i is correlated with variables x_{i-1} and x_{i+1}, if exists) and coefficients of the terms, $n-i$, are decreasing linearly by increasing the variable index number (i). As shown in Table 2, OMID does not work as well as TRMPS but works better than GA, CCEA-AVP, and DIRECT. Also, the difference between the average of TRMPS and OMID results is increasing with the increasing dimensionality. In 10-, 20-, and 30-dimensional problems, the differences between OMID and TRMPS results are 0.1347, 0.218, and 51.264, respectively. Two reasons are thought for why OMID results are worse than TRMPS in higher dimensional cases. First, when the number of variables increases, because of OMID setting a maximum of ten variables in a group, the number of correlations that are neglected increases and consequently the approximation error by decomposition increases. Second, the coefficient, $n-i$, is changing linearly, which means the difference between the intensities of correlations is not decreasing rapidly. In other words, the neglected correlations (i.e., the correlation of variables x_{11} – x_{30}) in the 30-dimensional case are not significantly less important than the kept correlations. To validate the assumption, a modified version of function SUR-T1-14 is designed and tested using the same algorithms. The structure of the function is the same as SUR-T1-14 with a difference in the coefficients of the terms of the summation; $(n-i)$ is changed to $[((n/2)-i)^2+1]$ (see the problem in the Appendix). The problem has two main differences from the original SUR-T1-14. First, the intensity of the correlations is more important when the variable indices are either very small or very large. The middle range variable correlations (e.g., correlation of x_{15} and x_{16} variables in 30-dimensional case) are less important. Second, the coefficient function is a second-order function rather than being linear. This will make the difference between the correlation intensities more significant than the original function.

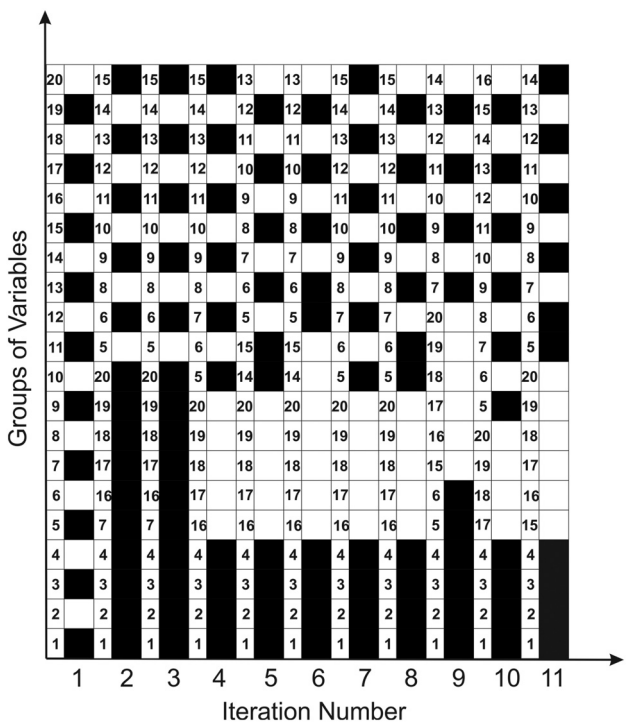

Fig. 3 Decomposed subproblems during OMID procedure for problem #10

Figure 3 shows the decomposed subproblems optimized by OMID during different iterations of a single run for the 20-dimensional modified SUR-T1-14 (problem #10). Again, black and white pattern fills are used for representing different groups of variables in subproblems with the corresponding variable indices to their left. The process starts with 20 independent variables being optimized at iteration 1. Iterations 2 and 3 assume a group of variables (x_1, x_2, x_3, x_4, x_7, x_{16}, x_{17}, x_{18}, x_{19}, x_{20}) and the rest of the variables are considered to be independent. It is clear that in the first few iterations, the decomposition result is not accurate because of not having enough sample points. However, the main group includes the most important variables. Iteration 4 resets the CCM matrix and uses the updated sample points for decomposition and as shown in the figure, two main groups of variables exist (x_1, x_2, x_3, x_4) and (x_{16}, x_{17}, x_{18}, x_{19}, x_{20}) and the rest are considered independent. As it continues, at iteration 10 after three resets of CCM, the groups are almost the same as the ones in iteration 4, which shows the importance of the grouped variable correlations in comparison with the rest of the correlations. The results show that OMID works better than TRMPS, GA, CCEA-AVP, and DIRECT in all 10-, 20-, and 30-dimensional modified cases. The results also validate our hypothesis that the poorer performance of OMID as compared to TRMPS on the original SUR-T1-14 is mostly due to the limitation of $d \leq 10$ for all subproblems set in OMID.

Problems 12–14 (Zakharov) have a structure with a combination of strong and weak correlations and also all the variables are correlated with each other. As shown in Table 2, OMID works better than TRMPS, GA, CCEA-AVP, and DIRECT in the 10-dimensional case but worse than them in 20-dimensional and 30-dimensional cases. The same two reasons for SUR-T1-14 should apply in this case as well. The problem again would have only one main group of variables and when the number of variables increases, the approximation error by decomposition increases because of neglecting an increasing number of correlations. Also, similar to SUR-T1-14, the coefficient which makes the correlation intensities different is changing linearly with respect to the variables' index. This property indicates that the intensities of the kept and omitted correlations are not much different. Another reason that can be thought for this case is the number of correlations existing in the structure. By looking at the structure, it can be seen that there are important (fourth order) correlations among every two, three, and four variables. For having maximum ten variables in a group, many important correlations would be omitted that can affect the accuracy of the optimization results.

The last function is Rosenbrock (problems 15–17) in which all the correlation intensities are important and in most of the cases, OMID could not find better results than TRMPS, GA, and DIRECT. However, OMID is found more efficient than the previously proposed metamodel-based decomposition method [34]. In the previous method [34], an excessive number of sample points would have been used to find if the problem is decomposable or not. Because OMID progressively reuses the optimization sample points, it does not waste any sample points and reaches a reasonable solution.

After testing with benchmark functions, OMID is applied to a real-world engineering problem. The problem is an assembly variation minimization problem, explained in the following section, Engineering Problem.

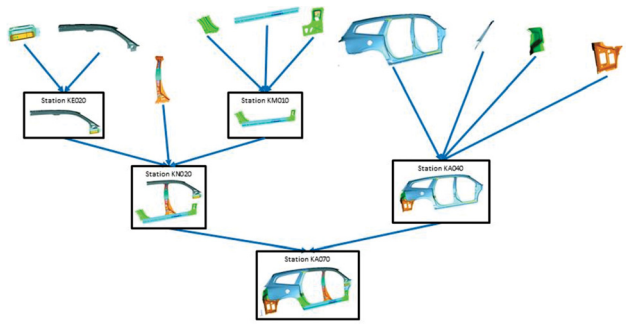

Fig. 4 Assembly sequence

Engineering Problem. The quality of sheet metal assemblies is affected by propagation of dimensional and geometric tolerances through the assembly process. Tolerance analysis shows the relationship between the tolerances of parts and the final assembly errors. The analysis indicates that with the same amount of tolerances of parts, the positions of the locators play important roles in the final assembly dimensional quality. The locators include four-way holes, two-way slots, and clamps that ensure the parts are positioned properly before welding. In addition to the part size tolerances, the locators themselves introduce new variations to the final assembly. The effects of locator variations on the final assembly dimensional variances can be minimized by selecting optimal positions of these locators.

The goal is to optimize the locators' positions for an auto-body assembly, as shown in Fig. 4. A dimensional analysis software tool, 3DCS ANALYST [44], is used to simulate the assembly process, and the final assembly quality measures are obtained for different configurations of the locators during the optimization process. The optimization algorithm is coded in MATLAB, which calls 3DCS to create an optimization loop. During the optimization process, the algorithm iteratively nominates new sets of locator configurations that are subsequently simulated in 3DCS. The overall assembly variation (i.e., the optimization objective function) is then returned to the optimizer, which suggests new configurations (design variable values).

A 3DCS design of experiments (DOE) script is used to send batches of experiments (or candidate configurations) from the optimizer (written in MATLAB) without launching the 3DCS graphical user interface. For optimization, hole, slot, and clamp locations are considered as design variables. Therefore, the number of optimization variables depends on the number of parts in the assembly and the number of locators on each part. Each part has one hole, one slot, and three clamps. With ten-parts, there are a total of 50 entities to be optimized. Furthermore, the location of each entity includes two coordinates in the assembly plane. Therefore, the optimization problem has 100 variables. As mentioned in the "Introduction" section, in the context of simulation-based design, usually a problem with dimensions more than ten ($d > 10$) is considered as a high-dimensional problem when the problem is computationally expensive. On the other hand, the objective function (i.e., variances at key product characteristic (KPC) points on the final assembly) of the optimization problem is calculated using 3DCS via Monte Carlo simulation. Thus, this optimization problem involves HEB functions.

In this problem, the goal is to minimize the final assembly error. The overall error is usually defined by a combination of measurements at KPC points on the assembly, whose variations from nominal values affect the final dimensional quality of the product. For the ten-part assembly (Fig. 4), 224 KPC points are specified on the assembly. The six-sigma values of the variation at the measure points from their nominal positions are calculated by 3DCS using Monte Carlo simulations. The objective function is a weighted sum of squares of the six-sigma values at all KPCs. In our studies, all KPC points are weighted equally. In practice, engineers can give more weight to some of the points if so desired. Two different case studies for optimization of the assembly station are presented

(1) The full 100-dimensional problem of a ten-part, using all the assembly stations.
(2) A 30-dimensional problem of a three-part assembly, using just one of the assembly stations.

The objective function is the same in both cases: it is the sum of squares of the six-sigma (6STD) of predefined measure points, which are distributed throughout the assembly. For the 30-variable case, 14 KPC points and for 100-variable case, 224 KPC points are defined. The optimization is performed using both TRMPS and OMID algorithms and the average of the ten runs is compared with the baseline objective function value before the optimization. In the 30-dimensional and 100-dimensional cases, 5000 and 10,000 sample points are used, respectively. Due to confidentiality, the results are shown relative to the baseline value by percentage. In 100-dimensional case, TRMPS reached a fixture layout with an objective function value equal to 54.96% of the baseline, while OMID reached to 19.7%. In the 30-dimensional case, TRMPS reached a fixture layout with an objective function equal to 81.25% of the baseline, while OMID reached to 71.88%. Therefore, OMID obtained more improvement in the objective function value comparing to the baseline value than TRMPS.

Conclusions

In this paper, a metamodeling-supported iterative decomposition strategy is proposed to optimize HEB problems. This approach, called OMID, starts with optimizing the input variables independently and iteratively builds metamodels, which are used to explore the intensities of variable correlations. PCA-HDMR is applied for metamodeling as it provides the ability to reuse the existing sample points generated during previous optimization iterations. The decomposition and optimization processes are intertwined to increase efficiency for global optimization of high-dimensional problems. The method is applied to 17 test problems with different dimensionalities from four different categories based on their inner variable structures and correlation intensities. The results of the method are compared with three different optimization methods, TRMPS, GA, CCEA-AVP, and DIRECT. The results showed that for problems with weak correlations and the problems with strong/weak combination of correlations, the proposed method is remarkably more accurate than other methods, with the same number of used sample points. For problems in which all correlations are strong or the intensities are close to each other, the method is not advantageous. However, because of integration of the decomposition and optimization phases, even if the problem is undecomposable, the proposed method still finds acceptable results, sometimes better than GA and DIRECT. A 100-dimensional assembly variation problem is optimized using the proposed method and the results are compared with TRMPS results and the baseline values. The results show significant improvement of OMID over TRMPS, with the same number of sample points. Future research will continue to improve the optimization strategy to solve for type 4 problems with all strong variable correlations.

Acknowledgment

The funding from the Natural Science and Engineering Research Council of Canada (CRDPJ421445-11) is gratefully acknowledged. Also, the authors would like to gratefully acknowledge the CCEA-AVP code provided by Dr. Singh and Dr. Ray at UNSW Australia.

Appendix

Name	D	Function	Variable ranges
SUR-T1-16	20	$f(\boldsymbol{x}) = \sum_{1}^{5}[(x_i + 10x_{i+5})^2 + 5(x_{i+10} - x_{i+15})^2 + (x_{i+5} - 2x_{i+10})^4 + 10(x_i - x_{i+15})^4]$	$-2 \le x_i \le 5$
F16	16	$f(x) = \sum_{i=1}^{16}\sum_{j=1}^{16} a_{ij}(x_i^2 + x_i + 1)(x_j^2 + x_j + 1)$ $a_{ij} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$-1 \le x_i \le 1$
PUR-T1-13	10, 20, 30	$f(\boldsymbol{x}) = \left[\sum_{i=1}^{n} i^3(x_i - 1)^2\right]^3$	$-3 \le x_i \le 3$
SUR-T1-14	10, 20, 30	$f(\boldsymbol{x}) = (x_1 - 1)^2 + (x_n - 1)^2 + n\sum_{i=1}^{n-1}(n - i)(x_i^2 - x_{i+1})^2$	$-3 \le x_i \le 2$
SUR-T1-14 modified	10, 20, 30	$f(\boldsymbol{x}) = (x_1 - 1)^2 + (x_n - 1)^2 + n\sum_{i=1}^{n-1}\left(\left(\frac{n}{2} - i\right)^2 + 1\right)(x_i^2 - x_{i+1})^2$	$-3 \le x_i \le 2$
Zakharov	10, 20, 30	$f(\boldsymbol{x}) = \sum_{1}^{n} x_i{}^2 + \left[\sum_{1}^{n}\frac{1}{2}\ i\ x_i\right]^2 + \left[\sum_{1}^{n}\frac{1}{2}\ i\ x_i\right]^4$	$-5 \le x_i \le 10$
Rosenbrock	10, 20, 30	$f(x) = \sum_{i=1}^{n-1}(100(x_{i+1} - x_i{}^2)^2 + (x_i - 1)^2)$	$-5 \le x_i \le 5$

References

[1] Shan, S., and Wang, G. G., 2010, "Survey of Modeling and Optimization Strategies to Solve High-Dimensional Design Problems With Computationally-Expensive Black-Box Functions," Struct. Multidiscip. Optim., **41**(2), pp. 219–241.

[2] Shan, S., and Wang, G. G., 2010, "Metamodeling for High Dimensional Simulation-Based Design Problems," ASME J. Mech. Des., **132**(5), p. 051009.

[3] Haftka, R. T., Scott, E. P., and Cruz, J. R., 1998, "Optimization and Experiments: A Survey," ASME Appl. Mech. Rev., **51**(7), pp. 435–448.

[4] Martins, J. R. R. A., Sturdza, P., and Alonso, J. J., 2003, "The Complex-Step Derivative Approximation," ACM Trans. Math. Software, **29**(3), pp. 245–262.

[5] Freund, J. B., 2010, "Adjoint-Based Optimization for Understanding and Suppressing Jet Noise," Procedia Eng., **6**, pp. 54–63.

[6] Mader, C. A., Martins, J. R. R. A., Alonso, J. J., and Van Der Weide, E., 2008, "ADjoint: An Approach for the Rapid Development of Discrete Adjoint Solvers," AIAA J., **46**(4), pp. 863–873.

[7] Bates, R. A., Buck, R. J., Riccomagno, E., and Wynn, H. P., 1996, "Experimental Design and Observation for Large Systems," J. R. Stat. Soc. Ser. B, **58**(1), pp. 77–94.

[8] Srivastava, A., Hacker, K., Lewis, K., and Simpson, T. W., 2004, "A Method for Using Legacy Data for Metamodel-Based Design of Large-Scale Systems," Struct. Multidiscip. Optim., **28**(2–3), pp. 146–155.

[9] Koch, P. N., Simpson, T. W., Allen, J. K., and Mistree, F., 1999, "Statistical Approximations for Multidisciplinary Design Optimization: The Problem of Size," Struct. Multidiscip. Optim., **36**(1), pp. 275–286.

[10] Myers, R. H., Montgomery, D. C., and Anderson-Cook, C. M., 2009, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, Wiley, New York.

[11] Ding, C., He, X., Zha, H., and Simon, H. D., 2002, "Adaptive Dimension Reduction for Clustering High Dimensional Data," International IEEE Conference on Data Mining (ICDM), Maebashi City, Japan, pp. 147–154.

[12] Kaya, H., Kaplan, M., and Saygın, H., 2004, "A Recursive Algorithm for Finding HDMR Terms for Sensitivity Analysis," Comput. Phys. Commun., **158**(2), pp. 106–112.

[13] Wang, G. G., Dong, Z., and Aitchison, P., 2001, "Adaptive Response Surface Method—A Global Optimization Scheme for Approximation-Based Design Problems," Eng. Optim., **33**(6), pp. 707–734.

[14] Wang, G. G., and Simpson, T., 2004, "Fuzzy Clustering Based Hierarchical Metamodeling for Design Space Reduction and Optimization," Eng. Optim., **36**(3), pp. 313–335.

[15] Winer, E. H., and Bloebaum, C. L., 2002, "Development of Visual Design Steering as an Aid in Large-Scale Multidisciplinary Design Optimization. Part I: Method Development," Struct. Multidiscip. Optim., **23**(6), pp. 412–424.

[16] Winer, E. H., and Bloebaum, C. L., 2002, "Development of Visual Design Steering as an Aid in Large-Scale Multidisciplinary Design Optimization. Part II: Method Validation," Struct. Multidiscip. Optim., **23**(6), pp. 425–435.

[17] Bloebaum, C., Hajela, P., and Sobieszczanski-Sobieski, J., 1992, "Non-Hierarchic System Decomposition in Structural Optimization," Eng. Optim., **19**(3), pp. 171–186.

[18] Kim, H. M., Michelena, N. F., Papalambros, P. Y., and Jiang, T., 2003, "Target Cascading in Optimal System Design," ASME J. Mech. Des., **125**(3), pp. 474–480.

[19] Michelena, N., Papalambros, P., Park, H., and Kulkarni, D., 1999, "Hierarchical Overlapping Coordination for Large-Scale Optimization by Decomposition," AIAA J., **37**(7), pp. 890–896.

[20] Allison, J. T., Kokkolaras, M., and Papalambros, P. Y., 2009, "Optimal Partitioning and Coordination Decisions in Decomposition-Based Design Optimization," ASME J. Mech. Des., **131**(8), p. 081008.

[21] Alexander, M. J., Allison, J. T., and Papalambros, P. Y., 2011, "Reduced Representations of Vector-Valued Coupling Variables in Decomposition-Based Design Optimization," Struct. Multidiscip. Optim., **44**(3), pp. 379–391.

[22] Liu, Y., Yao, X., Zhao, Q., and Higuchi, T., 2001, "Scaling Up Fast Evolutionary Programming With Cooperative Coevolution," 2001 Congress on Evolutionary Computation, Seoul, Korea, May 27–May30, pp. 1101–1108.

[23] Potter, M., and De Jong, K., 1994, "A Cooperative Coevolutionary Approach to Function Optimization," Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: Parallel Problem Solving from Nature (PPSN III), Y. Davidor, H.-P. Schwefel, and R. Männer, eds., Springer-Verlag, London, pp. 249–257.

[24] Shi, Y., Teng, H., and Li, Z., 2005, "Cooperative Co-Evolutionary Differential Evolution for Function Optimization," Advances in Natural Computation SE-147, L. Wang, K. Chen, and Y. Ong, eds., Springer, Berlin, Heidelberg, pp. 1080–1088.
[25] Yang, Z., Tang, K., and Yao, X., 2008, "Large Scale Evolutionary Optimization Using Cooperative Coevolution," Inf. Sci., **178**(15), pp. 2985–2999.
[26] Li, X., and Yao, X., 2012, "Cooperatively Coevolving Particle Swarms for Large Scale Optimization," IEEE Trans. Evol. Comput., **16**(2), pp. 210–224.
[27] Omidvar, M. N., Li, X., and Yao, X., 2011, "Smart Use of Computational Resources Based on Contribution for Cooperative Co-Evolutionary Algorithms," 13th Annual Conference on Genetic and Evolutionary Computation, Dublin, Ireland, pp. 1115–1122.
[28] Yang, Z., Tang, K., and Yao, X., 2008, "Multilevel Cooperative Coevolution for Large Scale Optimization," IEEE Congress on Evolutionary Computation, Hong Kong, June 1–6, pp. 1305–1312.
[29] Omidvar, M. N., Mei, Y., and Li, X., 2014, "Effective Decomposition of Large-Scale Separable Continuous Functions for Cooperative Co-Evolutionary Algorithms," 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, July 6–11, pp. 1305–1312.
[30] Omidvar, M. N., Li, X., and Mei, Y., 2014, "Cooperative Co-Evolution With Differential Grouping for Large Scale Optimization," IEEE Trans. Evol. Comput., **18**(3), pp. 378–393.
[31] Singh, H., and Ray, T., 2010, "Divide and Conquer in Coevolution: A Difficult Balancing Act," *Agent-Based Evolutionary Search SE-6*, R. Sarker and T. Ray, eds., Springer, Berlin, pp. 117–138.
[32] Mahdavi, S., Shiri, M. E., and Rahnamayan, S., 2014, "Cooperative Co-Evolution With a New Decomposition Method for Large-Scale," IEEE World Congress on Computational Intelligence, Beijing, July 6–11, pp. 1285–1292.
[33] Pirmoradi, Z., Haji Hajikolaei, K., and Wang, G. G., 2012, "Design Optimization on 'White-Box' Uncovered by Metamodeling," AIAA Paper No. 2012-1927.
[34] Hajikolaei, K. H., Pirmoradi, Z., Cheng, G. H., and Wang, G. G., 2014, "Decomposition for Large Scale Global Optimization Based on Quantified Variable Correlations Uncovered by Metamodeling," Eng. Optim., **47**(4), pp. 429–452.
[35] Sobol, I., 1993, "Sensitivity Estimates for Nonlinear Mathematical Models," Math. Model. Comput. Exp., **1**(4), pp. 407–414.
[36] Rabitz, H., and Alis, O. F., 1999, "General Foundations of High-Dimensional Model Representations," J. Math. Chem., **25**, pp. 197–233.
[37] Li, G., Rosenthal, C., and Rabitz, H., 2001, "High Dimensional Model Representations," J. Phys. Chem., **105**(33), pp. 7765–7777.
[38] Alı, Ö. F., and Rabitz, H., 2001, "Efficient Implementation of High Dimensional Model Representations," J. Math. Chem., **29**(2), pp. 127–142.
[39] Hajikolaei, K. H., and Wang, G. G., 2013, "High Dimensional Model Representation With Principal Component Analysis," ASME J. Mech. Des., **136**(1), p. 011003.
[40] Fu, J., and Wang, L., 2002, "A Random-Discretization Based Monte Carlo Sampling Method and Its Applications," Methodol. Comput. Appl. Probab., **4**(1), pp. 5–25.
[41] Hock, W., and Schittkowski, K., 1980, "Test Examples for Nonlinear Programming Codes," J. Optim. Theory Appl., **30**(1), pp. 127–129.
[42] Schittkowski, K., 1987, *More Test Examples for Nonlinear Programming Codes*, Springer-Verlag, New York.
[43] Cheng, G. H., Younis, A., Hajikolaei, K. H., and Wang, G. G., 2015, "Trust Region Based MPS Method for Global Optimization of High Dimensional Design Problems," ASME J. Mech. Des., **137**(2), p. 021407.
[44] "Dimensional Control Systems, Inc.," Last accessed 15 Sept., 2013, http://www.3dcs.com/