DRAFT

Proceedings of the ASME 2017 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2017
August 6-9, 2017, Cleveland, Ohio, USA

DETC2017-67976

ENHANCED GAUSSIAN PROCESS METAMODELING AND COLLABORATIVE OPTIMIZATION FOR VEHICLE SUSPENSION DESIGN OPTIMIZATION

Siyu Tao[1]
Graduate Research Assistant

Kohei Shintani[1]
Visiting Scholar, Ph.D.

Ramin Bostanabad[1]
Ph.D. Candidate

Yu-Chin Chan[1]
Graduate Research Assistant

Guang Yang[2]
Principal Engineer

Herb Meingast[2]
Manager

Wei Chen[1,*]
Professor

[1] Department of Mechanical Engineering, Northwestern University, Evanston, IL 60208, USA
[2] Toyota Motor North America, Inc., 1555 Woodridge Ave, Ann Arbor, MI 48105, USA
* Corresponding author, Email: weichen@northwestern.edu, URL: http://ideal.mech.northwestern.edu/

ABSTRACT

Dynamic stability is a key performance metric of motor vehicles and has a direct impact on passenger experience and customer satisfaction. The desired vehicle dynamics behavior can be achieved by optimizing the design of vehicle suspensions. Two challenges are associated with this design optimization task. The first one arises from the large number (e.g., 40 or 50) of design variables in modern suspension systems. Such multitude of variables not only makes it expensive to build a training dataset for metamodeling purposes, but also renders accurate surrogate modeling extremely difficult. The second challenge is a lack of guideline for choosing a proper multidisciplinary design optimization (MDO) method for a single MDO problem such as one for vehicle suspension design. In this paper, an enhanced Gaussian process (GP) metamodeling technique is developed and several versions of the collaborative optimization (CO) method are compared via a vehicle suspension design problem. In our enhanced GP modeling method, the model parameters are efficiently estimated using the smoothing effect of the so-called nugget parameter to reduce the search space. In addition, various versions of the CO method are studied where the enhanced collaborative optimization (ECO) method is found to perform the best. A simplified ECO formulation is also investigated to provide insights for future engineering applications.

1 INTRODUCTION

Vehicle dynamics greatly affects driving experience and customer satisfaction. The design of a vehicle system for achieving desired dynamic performance is challenging for two main reasons. First, the large number of design variables, coupled with the expensive computer simulations, makes it difficult to generate sufficient samples with design of experiments (DoE) for surrogate modeling. Even with a moderate number of samples, fitting a metamodel in such high dimensions is a daunting task. Under these circumstances, Gaussian processes (GP) provide a great choice of surrogate models as they are very flexible in terms of learning complex input-output maps. However, in our application to a vehicle suspension design problem with 46 design variables and 460 sample points for each response variable, no current GP modeling method in literature can fit accurate GP models within a manageable time. Second, it is challenging to choose a proper multidisciplinary design optimization (MDO) method for designing a vehicle system. Although a family of collaborative optimization (CO) [1-5] methods have been proposed to solve MDO problems in the past, their applications to a single MDO problem is not thoroughly investigated. The objective of this paper is to tackle the two challenges by developing an enhanced GP modeling technique and comparing different versions of the CO method via their application to a multidisciplinary vehicle suspension design problem.

GP models [6-10] are probably the most popular metamodels (aka surrogates, emulators, or response surfaces) used to replace expensive computer simulations. They have also been widely used in other applications, e.g., readily quantifying the prediction uncertainty [8, 9], helping identify the most

important inputs in the original computer model [11], and enabling tractable Bayesian calibration and bias correction [12-16]. However, as to be detailed in Section 2.1, the task of estimating the hyperparameters of a GP model becomes problematic when the number of design variables is very large. To resolve this issue, we develop an enhanced GP modeling method with a new two-stage hyperparameter estimation framework. In the first stage, with a relatively small computation cost, we effectively reduce the hyperparameter search space by leveraging the smoothing effect of a so-called nugget parameter. With the shrunk search space, the optimal hyperparameters can be efficiently found in the second stage. This enhanced method is under a general setting and readily applicable to any high-dimensional GP modeling tasks.

Among all the MDO methods developed in the past two decades, analytical target cascading (ATC) [17, 18] and collaborative optimization (CO) [1] are the two most popular ones applied to distributed engineering systems, with the former developed more specifically for hierarchical systems. ATC is well studied and has been successfully applied to vehicle suspension design problems that have hierarchical structures [19, 20]. CO has several major versions in literature, i.e., original CO [1], modified CO (MCO) [2, 3], and enhanced CO (ECO) [4, 5]. The MCO has very similar formulation as the original CO, but replaces L2-norms with L1-norms for all the consistency terms and transforms the system problem constraints into penalties. Compared with the previous two, ECO switches the roles of the system problem and the disciplinary sub-problems in terms of optimizing the global objective and enforcing the system consistency. It also introduces linearized nonlocal constraints in each sub-problem. While ECO manifests superior performance to its predecessors, it has more complex formulation especially due to the introduction of linearized nonlocal constraints. Therefore, in real applications, there is a potential trade-off between optimization performance and ease of implementation by choosing different versions of CO methods, which has not been well studied in literature. Herein we investigate these versions via a multidisciplinary vehicle suspension design problem. The goal of the design is to optimize the vehicle durability and riding comfort, which are quantified by the vehicle dynamic responses in simulations. Different road conditions under which the vehicle dynamics is simulated give rise to different analysis disciplines. Based on our comprehensive comparison between different versions of the CO method, the ECO formulation has the best performance and a simplified version of ECO is an alternative which compromises method performance for ease of implementation.

The remainder of the paper is organized as follows. The background on GP modeling along with our enhanced method is elaborated in Section 2. Details regarding three versions of the CO method and the theoretical comparison between them are provided in Section 3. The application of the enhanced GP modeling and different versions of the CO method to a vehicle suspension design problem is detailed in Section 4. Concluding remarks are provided in Section 5.

2 ENHANCED GAUSSIAN PROCESS (GP) METAMODELING METHOD

In this section, we first briefly describe the GP modeling procedure (see [7, 9, 21] for more details) and comment on the associated computational challenges. Then, with a $1D$ illustrative example, we elaborate on our technique to improve the efficiency of estimating the hyperparameters in GP models for problems with high dimensionality.

2.1 GP Modeling Procedure

Let $\boldsymbol{x} = [x_1, x_2, \dots, x_d]^T$ and $y(\boldsymbol{x})$ denote, respectively, the set of input variables over the d-dimensional input space $\mathbb{R}^d$ and the scalar response function. Suppose we observe $\boldsymbol{y} = [y(\boldsymbol{x}_1), y(\boldsymbol{x}_2), \dots, y(\boldsymbol{x}_n)]^T = [y_1, y_2, \dots, y_n]^T$ at n distinct locations $\boldsymbol{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$ and wish to predict $y(\boldsymbol{x}_0)$ for any $\boldsymbol{x}_0 \in \mathbb{R}^d$. The basic idea is to model $y(\boldsymbol{x})$ as a realization of a GP, denoted by $Y(\boldsymbol{x})$, with the parametric mean (expected value, $E[\cdot]$) and covariance functions ($c(\cdot,\cdot)$) being

$$E[Y(\mathbf{x})] = \mathbf{m}^T(\mathbf{x})\boldsymbol{\beta}, \tag{1}$$

$$c(\mathbf{x}, \mathbf{x}') = Cov[Y(\mathbf{x}), Y(\mathbf{x}')], \tag{2}$$

where $\boldsymbol{\beta}$ and $\boldsymbol{m}(\boldsymbol{x}) = [m_1(\boldsymbol{x}), m_2(\boldsymbol{x}), \dots, m_q(\boldsymbol{x})]^T$ are, respectively, vectors of unknown parameters and known basis functions (e.g., linear, quadratic, exponential). As for the covariance function, $c(\boldsymbol{x}, \boldsymbol{x}')$, the most common choice is

$$c(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left\{\sum_{i=1}^{d} -10^{\omega_i}\left(x_i - x_i'\right)^2\right\} = \sigma^2 r(\mathbf{x}, \mathbf{x}'), \tag{3}$$

where σ^2 is the variance of $Y(\boldsymbol{x})$, $r(\boldsymbol{x}, \boldsymbol{x}')$ is the Gaussian correlation function, and $\boldsymbol{\omega} = [\omega_1, \omega_2, \dots, \omega_d]^T$ are the correlation parameters of $r(\boldsymbol{x}, \boldsymbol{x}')$ that control the smoothness of the model (a large ω_i indicates a rough response surface along dimension i). $\boldsymbol{\beta}$, σ^2 and $\boldsymbol{\omega}$ are collectively called the hyperparameters, and the essential part of fitting a GP model is to estimate the hyperparameters via either the maximum likelihood estimation (MLE) or cross-validation (CV). Here, we use the MLE method as it is more robust and efficient [7].

In the MLE framework, the hyperparameters are estimated by maximizing the multivariate Gaussian likelihood function,

$$\begin{aligned}\left[\hat{\boldsymbol{\beta}}, \widehat{\sigma^2}, \hat{\boldsymbol{\omega}}\right] &= \arg\max_{\boldsymbol{\beta}, \sigma^2, \boldsymbol{\omega}} L[\boldsymbol{\beta}, \sigma^2, \boldsymbol{\omega} \mid \mathbf{y}] \\ &= \arg\max_{\boldsymbol{\beta}, \sigma^2, \boldsymbol{\omega}} [(2\pi\sigma^2)^{-\frac{n}{2}} |\mathbf{R}|^{-\frac{1}{2}} \\ &\quad \times \exp\{-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{M}\boldsymbol{\beta})^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{M}\boldsymbol{\beta})\}],\end{aligned} \tag{4}$$

or equivalently,

$$\begin{aligned}\left[\hat{\boldsymbol{\beta}}, \widehat{\sigma^2}, \hat{\boldsymbol{\omega}}\right] &= \arg\min_{\boldsymbol{\beta}, \sigma^2, \boldsymbol{\omega}} (-\log(L[\boldsymbol{\beta}, \sigma^2, \boldsymbol{\omega} \mid \mathbf{y}])) \\ &= \arg\min_{\boldsymbol{\beta}, \sigma^2, \boldsymbol{\omega}} [\frac{n}{2}\log(\sigma^2) + \frac{1}{2}\log(|\mathbf{R}|) \\ &\quad + \frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{M}\boldsymbol{\beta})^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{M}\boldsymbol{\beta})],\end{aligned} \tag{5}$$

where $\log(\cdot)$ is the natural logarithm, $\boldsymbol{M}$ is an $n \times q$ matrix with i^{th} row being $\boldsymbol{m}^T(\boldsymbol{x}_i)$, and $\boldsymbol{R}$ is an $n \times n$ matrix with $(i,j)^{th}$ element being $R_{ij} = r(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for $i, j = 1, \dots, n$. A

common approach for simplifying this optimization problem is to find $\boldsymbol{\beta}$ and σ^2 as functions of $\boldsymbol{\omega}$ to solve the optimization only in terms of $\boldsymbol{\omega}$. Setting the partial derivatives of the objective in Eqn. (5) with respect to $\boldsymbol{\beta}$ and σ^2 to zero yields

$$\hat{\boldsymbol{\beta}} = [\mathbf{M}^T\mathbf{R}^{-1}\mathbf{M}]^{-1}\mathbf{M}^T\mathbf{R}^{-1}\mathbf{y}, \tag{6}$$

$$\widehat{\sigma^2} = \frac{1}{n}\left(\mathbf{y}-\mathbf{M}\hat{\boldsymbol{\beta}}\right)^T \mathbf{R}^{-1}\left(\mathbf{y}-\mathbf{M}\hat{\boldsymbol{\beta}}\right). \tag{7}$$

Plugging these formulas ($\widehat{\boldsymbol{\beta}}$ and $\widehat{\sigma^2}$ are functions of $\boldsymbol{\omega}$ as R_{ij} depends on $\boldsymbol{\omega}$) into Eqn. (5) yields (constants dropped)

$$\hat{\boldsymbol{\omega}} = \arg\min_{\boldsymbol{\omega}}\left(n\log\left(\widehat{\sigma^2}\right)+\log\left(|\mathbf{R}|\right)\right) = \arg\min_{\boldsymbol{\omega}} L. \tag{8}$$

By numerically solving Eqn. (8), one can find $\hat{\boldsymbol{\omega}}$. Then $\widehat{\boldsymbol{\beta}}$ and $\widehat{\sigma^2}$ can be obtained via Eqn. (6) and Eqn. (7). Once the hyperparameters are estimated, the GP prediction for $y(\boldsymbol{x}_0)$ and the associated mean squared error (MSE) can be calculated with

$$\hat{y}(\mathbf{x}_0) = \mathbf{m}^T(\mathbf{x}_0)\hat{\boldsymbol{\beta}} + \mathbf{r}^T(\mathbf{x}_0)\mathbf{R}^{-1}(\mathbf{y}-\mathbf{M}\hat{\boldsymbol{\beta}}), \tag{9}$$

$$\begin{aligned} MSE[\hat{y}(\mathbf{x}_0)] = \widehat{\sigma^2}[r(\mathbf{x}_0,\mathbf{x}_0) - \mathbf{r}^T(\mathbf{x}_0)\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}_0) \\ +\mathbf{W}^T(\mathbf{M}^T\mathbf{R}^{-1}\mathbf{M})^{-1}\mathbf{W}], \end{aligned} \tag{10}$$

where $\boldsymbol{W} = \boldsymbol{m}(\boldsymbol{x}_0) - \boldsymbol{M}^T\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}_0)$ and $\boldsymbol{r}(\boldsymbol{x}_0)$ is an $n\times 1$ vector whose i^{th} element is $r(\boldsymbol{x}_i, \boldsymbol{x}_0)$. Since in this work we use the GP model only for interpolation, reversion to the mean is of no concern. Hence, we use a constant prior mean $m(\boldsymbol{x}) = 1$.

The heart of fitting a GP model lies in efficient optimization of L in Eqn. (8). This is especially challenging for problems with high dimensionality, i.e., large number of input variables. It has been shown [22-24] that L usually has a complex profile with multiple local optima. Previous works have used gradient-based techniques where, to ensure global optimality, the optimization is run multiple times from different starting points. The downside of using such an approach is that as the problem dimensionality increases, the number of different starting points must be increased in proportion to the volume of the search space to ensure global optimality of the solution. This issue is exacerbated as the number of training samples increases because the computational cost of evaluating $\boldsymbol{R}^{-1}$ in Eqn. (7) and subsequently L and the entire optimization process increases.

2.2 New Two-stage Parameter Estimation Technique

Having in mind that the optimization efficiency for determining GP hyperparameters can deteriorate greatly due to the drastic increase of the required number of optimization runs as the dimensionality of $\boldsymbol{\omega}$ escalates, we now detail our approach to this issue via a two-stage parameter estimation technique. The ω_i's in Eqn. (3) can take any real number in $(-\infty, \infty)$, but previous research [24, 25] has shown that for most GP models $\omega_i \in [-10, 10]$. Thus, the q initial points of the gradient-based optimization process may be chosen via a space-filling algorithm (e.g., optimal Latin hypercube sampling [26]) in the $[-10, 10]^d$ hypercube. Our essential idea is to use the smoothing functionality of the so-called nugget parameter, δ, to shrink this search space and thus require a smaller q.

The nugget parameter is usually a very small number, e.g., 10^{-8}, that was originally proposed to address (i) noisy observations and/or (ii) the potential numerical difficulty in inverting the covariance matrix, $\boldsymbol{R}$. Case (ii) usually happens when the training dataset is dense (i.e., n is large). In both cases, $\boldsymbol{R}$ in all the equations in Section 2.1 is replaced with

$$\mathbf{R}_\delta = \mathbf{R} + \delta\mathbf{I}, \tag{11}$$

where $\boldsymbol{I}$ is the $n\times n$ identity matrix. In case (i), δ is generally set (or estimated) to be the variance of the Gaussian noise [7, 10]. In case (ii), δ is used only if $\boldsymbol{R}$ is ill-conditioned. A typical formula for δ is

$$\delta = \begin{cases} e_{th} - e_s & \text{if } e_s < e_{th} \\ 0 & \text{otherwise} \end{cases}, \tag{12}$$

where e_s is the smallest eigenvalue of $\boldsymbol{R}$, and e_{th} is the smallest eigenvalue that $\boldsymbol{R}_{\boldsymbol{\delta}}$ can take to avoid numerical errors (e_{th} depends on machine precision and generally $e_{th}\sim 10^{-8}$).

Our use of the nugget parameter is based on an entirely different reason. Our studies indicate that choosing a very large value for e_{th} such as 10^{-1} *smooths out* most of the local optima in the profile of L while slightly shifting the location of the global optimum. As this constraint is slightly loosened (e.g., $e_{th} = 10^{-2}$), the location of the global optimum for $\boldsymbol{\omega}$ in Eqn. (8) closes on to that of the *true* global optimum (when no or a very small δ is used). We use these observations to estimate the location of the *true* global optimum as follows:

Step 1: Bound the search space to $[a, b]^d$ and find optimal solution $\boldsymbol{\omega}^s$ by solving the optimization problem in Eqn. (8) a few times (e.g., 3 or 5 times) while setting e_{th} to a large value (e.g., 10^{-3}) in Eqn. (12).

Step 2: Refine the lower ($\boldsymbol{L}$) and upper bound ($\boldsymbol{U}$) of the search space along each direction via $\boldsymbol{\omega}^s$ as:

```
For i = 1:d
    if ω^s(i) == a
        L(i) = −10
        U(i) = 0
    else if ω^s(i) == b
        L(i) = 0
        U(i) = 3
    else
        L(i) = ω^s(i) − 1
        U(i) = ω^s(i) + 1
end
```

Once the range is estimated, Eqn. (8) is solved again but with a very small e_{th} to find the *true* global optimum. Having the search space shrunk, one will need to solve Eqn. (8) only a few times (each time starting from a different point) rather than a lot more times, and the reduction factor of the number of the times is typically that of the search space volume. Regarding the "for loop" in **Step 2** we note that:

1. If including x_i in the GP model does not increase its predictive power (i.e., $y(\boldsymbol{x})$ is invariant regarding x_i), the global optimum along dimension i will be located at the lower bound of the initial search space (i.e., $\boldsymbol{\omega}^s(i) == a$) in **Stage 1**. In this case, $\boldsymbol{L}(i)$ in **Stage 2** is set to -10 to allow the GP emulator render x_i as unimportant as it needs to be.

2. If $y(\boldsymbol{x})$ is rough (i.e., changes rapidly) along the i^{th} dimension, the global optimum in that direction will be located at the upper bound of the initial search space (i.e., $\boldsymbol{\omega}^s(i) == b$) in **Stage 2**. In this case, $\boldsymbol{U}(i)$ in **Stage 2** is set to $b+1$ to allow the GP emulator become as rough as required along x_i. Here, as opposed to the above, the range is increased only by a unit because $\boldsymbol{\omega}$ appears as the exponent in Eqn. (3).

3. The last two conditions in the loop are rather conservative: While significantly reducing the range along each dimension from $[-10, 10]$, we have made sure the range is still wide enough to account for the slight shift in the location of the global solution (see Figure 1 (a) as an example).

4. Based on our empirical studies, setting $[a, b] = [-3, 2]$ works very well for a wide range of problems, but most parameters in the method including $[a, b]$ can be adjusted as needed, and the mechanism of the algorithm ensures the search result has low sensitivity to the choice of $[a, b]$.

The above procedure is illustrated in Figure 1 via a $1D$ function with 11 training inputs located at $-1, -0.8, \ldots, 0.8, 1$

$$y(x) = \sin(5x) + x\log(x+1.1). \tag{13}$$

As can be observed in Figure 1 (a) and (b), the profile of L is complex and if one were to solve Eqn. (8) only once, a local optimum might be incorrectly chosen for $\hat{\omega}$. It is evident, however, that imposing an overly restrictive constraint on e_s results in a smooth profile where the global optimum can be found much more easily (the red curves in Figure 1 (a) and (b)).

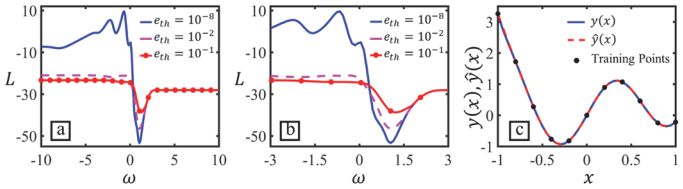

Figure 1. (a) The profile of L as a function of ω for three different values of e_{th}. (b) Same figure as (a) but over a smaller range for ω. (c) The exact function and the fitted GP model to the training points.

We note that the described procedure is not only general, but most advantageous for high-dimensional ones where one cannot afford solving Eqn. (8) many times. In our design problem to be demonstrated in Section 4, five GP models are fitted; each with 460 samples in 46 dimensions. Using the technique detailed in this section, we are able to fit accurate models in less than an hour on an ordinary desktop (Intel® Core™ i7-6700 at 3.40GHz with 4 cores). As to be detailed in Sections 4.2 and 4.5, the accuracy of the fitted models is quite high.

3 COLLABORATIVE OPTIMIZATION METHODS

3.1 Original Collaborative Optimization Method

Collaborative optimization (CO) method is an MDO method initially developed for large-scale distributed systems [1] and commonly applied to non-hierarchical ones. To understand the CO method, we first formulate a general MDO problem as

$$\begin{aligned} \min \quad & F(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{y}), \\ w.r.t. \quad & \mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N, \hat{\mathbf{y}}, \\ s.t. \quad & \mathbf{c}_0(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{y}) \geq 0, \\ & \mathbf{c}_i\left(\mathbf{x}_0, \mathbf{x}_i, \mathbf{y}_i\left(\mathbf{x}_0, \mathbf{x}_i, \mathbf{y}_{j\neq i}\right)\right) \geq 0, \\ & \mathbf{c}_i^c = \hat{\mathbf{y}}_i - \mathbf{y}_i = 0, \end{aligned} \tag{14}$$

where notations are adopted from [27] and listed in Table 1. Note that this general problem (14) is slightly simplified compared with that in [27] because the purpose here is for explaining only the CO architecture. To use the CO method, the original problem Eqn. (14) is decomposed into N disciplinary sub-problems with a system-level problem created to coordinate the optimization processes across all the sub-problems. The general architecture of the original CO method is illustrated in Figure 2. The purpose of having copies (denoted by hats) of variables is to facilitate the description of the interdisciplinary coordination process. Readers may refer to [27] for more details.

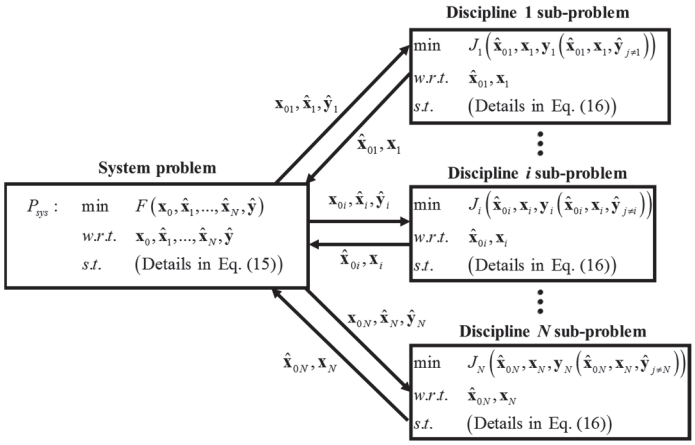

Figure 2. CO formulation for solving MDO problems

Table 1. Nomenclature for MDO problems and CO formulations

Variable	Description
N	Number of disciplines
$F(\cdot)$	Global objective function
$\boldsymbol{c}_0(\cdot)$	Constraint functions analyzed by the system problem solver
$\boldsymbol{c}_i(\cdot)$	Local constraint functions in Discipline i
$\boldsymbol{c}_c(\cdot)$	Consistency constraint functions
$\boldsymbol{x}_0$	Shared design variables across at least two different disciplines
$\boldsymbol{x}_i$	Local design variables of Discipline i
$\boldsymbol{y}_i(\boldsymbol{x}_0, \boldsymbol{x}_i, \boldsymbol{y}_{j\neq i})$	Analysis responses (functions of local design variables, shared design variables, and/or responses from other disciplines) of Discipline i
$\boldsymbol{y} = [\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_N^T]^T$	Collection of responses from all the discipline analyses
$\hat{\boldsymbol{x}}_{0i}$	Copy of shared design variables stored in Discipline i
$\hat{\boldsymbol{x}}_i$	Copy of local design variables of Discipline i
$\hat{\boldsymbol{y}} = [\hat{\boldsymbol{y}}_1^T, \ldots, \hat{\boldsymbol{y}}_N^T]^T$	Copy of responses from all the discipline analyses

In the CO architecture shown in Figure 2, the system optimization problem is

$$\begin{aligned} \min \quad & F(\mathbf{x}_0, \hat{\mathbf{x}}_1, ..., \hat{\mathbf{x}}_N, \hat{\mathbf{y}}), \\ w.r.t. \quad & \mathbf{x}_0, \hat{\mathbf{x}}_1, ..., \hat{\mathbf{x}}_N, \hat{\mathbf{y}}, \\ s.t. \quad & \mathbf{c}_0(\mathbf{x}_0, \hat{\mathbf{x}}_1, ..., \hat{\mathbf{x}}_N, \hat{\mathbf{y}}) \geq \mathbf{0}, \\ & J_i^* = \left\| \hat{\mathbf{x}}_{0i} - \mathbf{x}_0 \right\|_2^2 + \left\| \hat{\mathbf{x}}_i - \mathbf{x}_i \right\|_2^2 \\ & \quad + \left\| \hat{\mathbf{y}}_i - \mathbf{y}_i(\hat{\mathbf{x}}_{0i}, \mathbf{x}_i, \hat{\mathbf{y}}_{j \neq i}) \right\|_2^2 = 0, \quad i = \{1, 2, ..., N\}. \end{aligned} \tag{15}$$

The discipline i sub-problem is

$$\begin{aligned} \min \quad & J_i(\hat{\mathbf{x}}_{0i}, \mathbf{x}_i, \mathbf{y}_i(\hat{\mathbf{x}}_{0i}, \mathbf{x}_i, \hat{\mathbf{y}}_{j \neq i})), \\ w.r.t. \quad & \hat{\mathbf{x}}_{0i}, \mathbf{x}_i, \\ s.t. \quad & \mathbf{c}_i(\hat{\mathbf{x}}_{0i}, \mathbf{x}_i, \mathbf{y}_i(\hat{\mathbf{x}}_{0i}, \mathbf{x}_i, \hat{\mathbf{y}}_{j \neq i})) \geq \mathbf{0}, \end{aligned} \tag{16}$$

where $J_i(\hat{\boldsymbol{x}}_{0i}, \boldsymbol{x}_i, \boldsymbol{y}_i(\hat{\boldsymbol{x}}_{0i}, \boldsymbol{x}_i, \hat{\boldsymbol{y}}_{j \neq i}))$ has the same form as J_i^* in Eqn. (15) but different independent variables. According to these formulations, the system optimization problem is responsible for minimizing the global objective function and ensuring the consistency between disciplinary design variables ($\hat{\boldsymbol{x}}_{0i}$, $\boldsymbol{x}_i$) and analysis responses ($\boldsymbol{y}_i$) by coming up with a set of common targets ($\boldsymbol{x}_{0i}$, $\hat{\boldsymbol{x}}_i$, and $\hat{\boldsymbol{y}}_i$) for them. Meanwhile, the sub-problem solvers minimize the inconsistency between the set of targets ($\boldsymbol{x}_{0i}$, $\hat{\boldsymbol{x}}_i$, and $\hat{\boldsymbol{y}}_i$) from the system solver and the local design variables and responses ($\hat{\boldsymbol{x}}_{0i}$, $\boldsymbol{x}_i$, and $\boldsymbol{y}_i$).

The original CO method has the advantage of fully separating the discipline analyses and manifests good performance in some applications such as an aerospace vehicle design problem [1]. However, several numerical problems are faced when it is coupled with gradient-based optimization algorithms [2, 28]. Five major problems are summarized in [29]:

1. The system problem constraints J_i^* are generally non-smooth functions and therefore non-differentiable.
2. The Jacobian for the system problem constraints is singular at the optimum.
3. Several local minima may exist in sub-problems for each set of target values assigned by the system problem solver.
4. The Lagrange multipliers in the disciplinary sub-problems are zero or converge to zero at optimum.
5. The system problem has no information about the active constraints in the sub-problems.

Attempts are made to modify the CO method to address these difficulties. Two major improved versions, modified collaborative optimization (MCO) and enhanced collaborative optimization (ECO) are introduced in subsections 3.2 and 3.3.

3.2 Modified Collaborative Optimization Method

In the modified collaborative optimization (MCO) [2], the formulations of the system problem and disciplinary sub-problems are slightly modified compared with the original CO. The system optimization problem is changed to

$$\begin{aligned} \min \quad & F(\mathbf{x}_0, \hat{\mathbf{x}}_1, ..., \hat{\mathbf{x}}_N, \hat{\mathbf{y}}) + \alpha \sum_{i=1}^{N} J_i^*, \\ w.r.t. \quad & \mathbf{x}_0, \hat{\mathbf{x}}_1, ..., \hat{\mathbf{x}}_N, \hat{\mathbf{y}}, \\ s.t. \quad & \mathbf{c}_0(\mathbf{x}_0, \hat{\mathbf{x}}_1, ..., \hat{\mathbf{x}}_N, \hat{\mathbf{y}}) \geq \mathbf{0}, \end{aligned} \tag{17}$$

where α is a positive penalty parameter and $J_i^* = \|\hat{\boldsymbol{x}}_{0i} - \boldsymbol{x}_0\|_1 + \|\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i\|_1 + \left\|\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i(\hat{\boldsymbol{x}}_{0i}, \boldsymbol{x}_i, \hat{\boldsymbol{y}}_{j \neq i})\right\|_1$. The discipline i sub-problem is modified as

$$\begin{aligned} \min \quad & J_i(\hat{\mathbf{x}}_{0i}, \mathbf{x}_i, \mathbf{y}_i(\hat{\mathbf{x}}_{0i}, \mathbf{x}_i, \hat{\mathbf{y}}_{j \neq i})), \\ w.r.t. \quad & \hat{\mathbf{x}}_{0i}, \mathbf{x}_i, \\ s.t. \quad & \mathbf{c}_i(\hat{\mathbf{x}}_{0i}, \mathbf{x}_i, \mathbf{y}_i(\hat{\mathbf{x}}_{0i}, \mathbf{x}_i, \hat{\mathbf{y}}_{j \neq i})) \geq \mathbf{0}. \end{aligned} \tag{18}$$

where $J_i(\hat{\boldsymbol{x}}_{0i}, \boldsymbol{x}_i, \boldsymbol{y}_i(\hat{\boldsymbol{x}}_{0i}, \boldsymbol{x}_i, \hat{\boldsymbol{y}}_{j \neq i}))$ has the same form as J_i^* in Eqn. (17) but different independent variables.

MCO overcomes at least three of the five difficulties associated with the original CO listed in the previous subsection. Issues 2 and 4 are addressed by using L1-norm instead of L2-norm in the sub-problem objective functions and integrating the equality constraints in the system problem as penalty terms into the objective function. Issue 1 is tackled by solving a sequence of the so-called perturbed MCO problems. However, these perturbed problems bring about the ill-conditioning issue due to the use of barrier functions and an adequate update rule for the barrier parameter is not provided in literature. Also considering that the complexity of solving the perturbed problems is impractical for industrial applications, we adopt a simplified MCO method where Equations (17) and (18) are directly solved, and test it with the MDO problem in Section 4.

3.3 Enhanced Collaborative Optimization Method

The most recent version of CO is the enhanced collaborative optimization (ECO) [4, 5] method which has largely different formulation from that of the original CO (see Figure 3).

The system optimization problem for ECO is

$$\begin{aligned} \min \quad & J_0 = \sum_{i=1}^{N} \left(\left\| \hat{\mathbf{x}}_{0i} - \mathbf{x}_0 \right\|_2^2 + \left\| \hat{\mathbf{y}}_i - \mathbf{y}_i(\mathbf{x}_0, \mathbf{x}_i, \hat{\mathbf{y}}_{j \neq i}) \right\|_2^2 \right), \\ w.r.t. \quad & \mathbf{x}_0, \hat{\mathbf{y}}. \end{aligned} \tag{19}$$

The discipline i sub-problem is

$$\begin{aligned} \min \quad & J_i = \tilde{F}_0(\hat{\mathbf{x}}_{0i}, \mathbf{y}_i(\hat{\mathbf{x}}_{0i}, \mathbf{x}_i, \hat{\mathbf{y}}_{j \neq i})) \\ & \quad + w_{Ci} \left(\left\| \hat{\mathbf{x}}_{0i} - \mathbf{x}_0 \right\|_2^2 + \left\| \hat{\mathbf{y}}_i - \mathbf{y}_i(\hat{\mathbf{x}}_{0i}, \mathbf{x}_i, \hat{\mathbf{y}}_{j \neq i}) \right\|_2^2 \right) \\ & \quad + w_{Fi} \sum_{j=1, j \neq i}^{N} \sum_{k=1}^{n_s} s_{jk}, \\ w.r.t. \quad & \hat{\mathbf{x}}_{0i}, \mathbf{x}_i, \mathbf{s}_{j \neq i}, \\ s.t. \quad & \mathbf{c}_i(\hat{\mathbf{x}}_{0i}, \mathbf{x}_i, \mathbf{y}_i(\hat{\mathbf{x}}_{0i}, \mathbf{x}_i, \hat{\mathbf{y}}_{j \neq i})) \geq \mathbf{0}, \\ & \tilde{\mathbf{c}}_j(\hat{\mathbf{x}}_{0i}) + \mathbf{s}_j \geq \mathbf{0}, \quad j = \{1, ..., i-1, i+1, ..., N\}, \\ & \mathbf{s}_j \geq \mathbf{0}, \quad j = \{1, ..., i-1, i+1, ..., N\}, \end{aligned} \tag{20}$$

where in addition to the notations listed in Table 1, $\tilde{F}_0(\cdot)$ is a quadratic model of the global objective function, $\tilde{\boldsymbol{c}}_j(\cdot)$ is the linearized model of the constraint function $\boldsymbol{c}_j(\cdot)$ from discipline j ($j \neq i$), $\boldsymbol{s}_j$ is the vector of slack variables for the linearized constraint function $\tilde{\boldsymbol{c}}_j(\cdot)$, and w_{Ci} and w_{Fi} are weights for the consistency and slack variable penalties, respectively. To ensure optimality, w_{Fi} must be larger than the largest Lagrange multiplier for the discipline i sub-problem. w_{Ci} serve to guide the optimization toward a consistent solution [27]. In principle,

each w_{Ci} must be driven to infinity to enforce exact consistency, but in practice finite values are used to balance consistency and exploration of a larger design space [4].

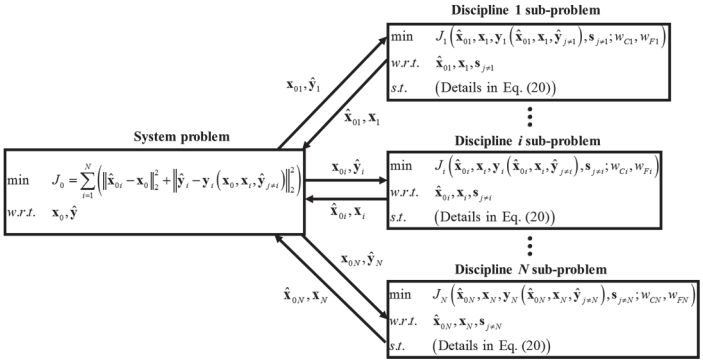

Figure 3. ECO formulation for solving MDO problems

The formulation of ECO is radically different from that of the original CO and theoretically eliminates almost all the five major problems introduced in Section 3.1. *A distinct feature of ECO is that the roles of the system problem and the disciplinary sub-problems are inverted compared with other versions of CO*; the system problem only handles the variables and responses consistency between different disciplinary groups while the sub-problems minimize the global objective function. Another feature of ECO lies in the introduction of linear models of nonlocal constraints (represented by $\tilde{\boldsymbol{c}}_j(\cdot)$ $(j \neq i)$ in discipline i sub-problem) and quadratic models of the global objective function (represented by $\tilde{F}_0(\cdot)$ in each subsystem-level problem). In this way, not only issues 1, 2 and 4 listed in Section 3.1 are addressed, but issues 3 and 5 are also alleviated since $\tilde{\boldsymbol{c}}_j(\cdot)$ $(j \neq i)$ and $\tilde{F}_0(\cdot)$ in discipline i respectively increase each discipline's "awareness" of the constraints in other disciplines and enhance, if not guarantee, the uniqueness of the solutions to subsystem-level problems. Although such features necessitate more function evaluations and some interdisciplinary data communication (incurred by building linear and quadratic approximations of nonlocal constraints and objectives), they preserve a high level of interdisciplinary analysis independence and provide ECO with a better convergence rate.

It should be noted that due to the nature of our application problem, in this paper the ECO method is applied slightly differently. First, when the system problem and disciplinary sub-problems are solved, gradient-based algorithms are used to search for only one local optimum rather than the global one with the starting point being the local optimum found at the previous iteration. The reason is that all the disciplinary sub-problems have large design space (caused by the large number of design variables), hence searching for the global optimum is costly. Second, as a by-product of the above modification, the slack variables in each discipline are not effective in improving the performance of the method. Hence, all the slack variables are dropped, which is equivalent to setting the penalty parameters for slack variables to infinity. Finally, in addition to the original ECO method, a simplified ECO method where the linearized constraints $\tilde{\boldsymbol{c}}_j(\cdot)$ $(j \neq i)$ are dropped in every discipline i sub-problem is proposed for ease and efficiency of implementation. Both original and simplified ECO methods are applied to the vehicle suspension MDO problem in Section 4 where insights are drawn from the comparison between the two methods.

4 APPLICATION TO A VEHICLE SUSPENSION PARAMETER DESIGN PROBLEM

4.1 General Problem Description

The enhanced GP metamodeling and different versions of the CO method introduced in Sections 2 and 3 are applied to a suspension parameter design problem for vehicle dynamics. The multi-body dynamic vehicle system used in the problem is developed from an Altair® MotionView® sample vehicle model. It contains a front MacPherson suspension and a rear strut suspension subsystem. Mechanical property parameters such as stiffness and damping coefficients of several components of the suspensions are selected as design variables. Figure 4 illustrates the structure of the suspension in a vehicle body. Figure 5 and Figure 6 show the components of the front and rear suspensions. All the components and associated design variables (of which there are 46 in total) are listed in Table 3. The goal of this design problem is to optimize the vehicle durability while improving its riding comfort with at least 10% improvement of the performance indicators. Therefore, three multi-body dynamics simulation events (rough road, brake and double-lane change) that represent vehicle handling conditions are conducted, and five key indicator responses (listed in Table 2 and illustrated in Figure 4) of the vehicle durability and riding comfort are extracted from the simulations. These extracted response values are their maximum variation amplitudes over a certain time period during the simulations. For simplicity, we will refer to them by their variable names, $f_1(\boldsymbol{x}_0)$ through $f_5(\boldsymbol{x}_0)$, defined in Table 2. Notice that all five responses are functions of a single set of shared variables $\boldsymbol{x}_0$, which stands for the complete set of 46 design variables. Since all the dynamics simulations use the same vehicle suspension model, all response analyses and hence disciplines (to be defined shortly) share the same set of design variables.

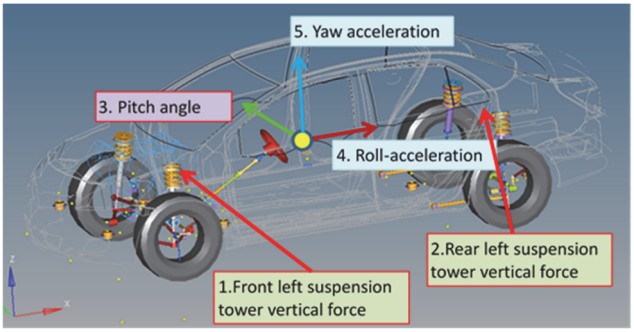

Figure 4. Vehicle body and suspension model with five dynamical responses

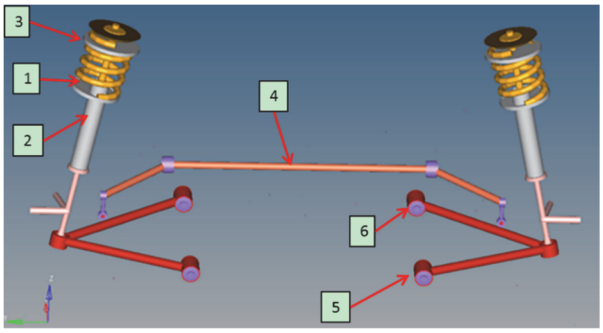

Figure 5. Front suspension design components (numbers in accordance with Table 2)

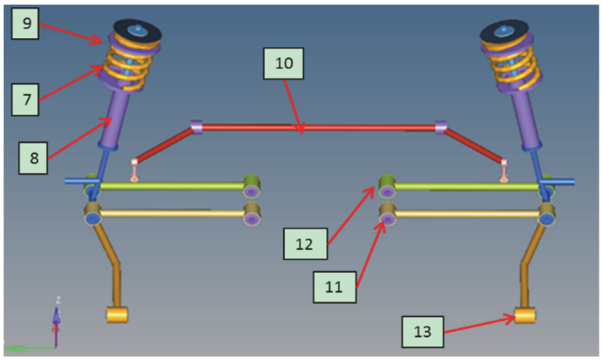

Figure 6. Rear suspension design components (numbers in accordance with Table 2)

Table 3. All the suspension components and design variables

Index	Component name	Number of design variable(s)	Design variable name(s)
1	Front coil spring	1	x_1
2	Front damper	5	x_2 to x_6
3	Front bump stopper	1	x_7
4	Front stabilizer bar	1	x_8
5	Front lower control arm front bushing	6	x_9 to x_{14}
6	Front lower control arm rear bushing	6	x_{15} to x_{20}
7	Rear coil spring	1	x_{21}
8	Rear damper	5	x_{22} to x_{26}
9	Rear bump stopper	1	x_{27}
10	Rear stabilizer bar	1	x_{28}
11	Rear suspension front link	6	x_{29} to x_{34}
12	Rear suspension rear link	6	x_{35} to x_{40}
13	Trailing arm bushing	6	x_{41} to x_{46}

The formal MDO formulation for the design problem described above is

$$\begin{aligned} &\min && f_1(\mathbf{x}_0), \\ &w.r.t. && \mathbf{x}_0, \\ &s.t. && \mathbf{c}_1(\mathbf{x}_0)=\begin{bmatrix}0.9-f_2(\mathbf{x}_0)\\0.9-f_3(\mathbf{x}_0)\end{bmatrix}\geq 0, \\ & && \mathbf{c}_2(\mathbf{x}_0)=\begin{bmatrix}0.9-f_4(\mathbf{x}_0)\\0.9-f_5(\mathbf{x}_0)\end{bmatrix}\geq 0, \\ & && 0\leq \mathbf{x}_0 \leq 1, \end{aligned} \tag{21}$$

where the ranges of all design variables, $\boldsymbol{x}_0$, are normalized to the unit interval $[0,1]$ and the values of all response variables, $f_1(\boldsymbol{x}_0)$ through $f_5(\boldsymbol{x}_0)$, are normalized against their baseline values so that the baseline design is $\boldsymbol{x}_{0,\text{base}} = 0.5$ and $f_i(\boldsymbol{x}_{0,\text{base}}) = 1.0$ for $i = 1,2,\ldots,5$. Even though there is no clear distinction of disciplines, for the purpose of this study, the "disciplinary" structure of this MDO problem is artificially created as shown in Figure 7.

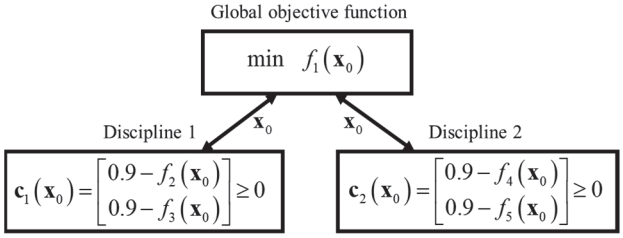

Figure 7. Multidisciplinary structure of the vehicle suspension MDO problem

The disciplinary assignment is justified as follows. First, per the distinction of the three simulation events, the two responses that evaluate double lane change performance, $f_4(\boldsymbol{x}_0)$ and $f_5(\boldsymbol{x}_0)$, are grouped under Discipline 2. Next, we look at the only two vehicle durability indictors, $f_1(\boldsymbol{x}_0)$ and $f_2(\boldsymbol{x}_0)$, and choose the former as the global objective because it is considered more critical. Additionally, considering that $f_1(\boldsymbol{x}_0)$ and $f_2(\boldsymbol{x}_0)$ compete with each other, to test the CO methods' ability to handle the competition between minimizing the global objective and satisfying the constraints in disciplinary sub-problems, we separate $f_2(\boldsymbol{x}_0)$ from $f_1(\boldsymbol{x}_0)$ to induce more competition and group it with $f_3(\boldsymbol{x}_0)$ (an indicator of brake performance) under Discipline 1. Overall, we have a balanced number of constraints in each disciplinary sub-problem. As to be seen in Section 4.4, the above "artificial" disciplinary formulation allows us to differentiate the performance of different CO methods.

Table 2. All the simulation events and corresponding response variables

Simulation event	Response name	Response variable name*
Rough road (by four-post test)	Front left suspension tower vertical load	$f_1(\boldsymbol{x}_0)$
	Rear left suspension tower vertical load	$f_2(\boldsymbol{x}_0)$
Brake	Vehicle center of gravity (CG) pitch angle	$f_3(\boldsymbol{x}_0)$
ISO double lane change	Vehicle CG roll angular acceleration	$f_4(\boldsymbol{x}_0)$
	Vehicle CG yaw angular acceleration	$f_5(\boldsymbol{x}_0)$

* $\boldsymbol{x}_0$ stands for the vector of all the 46 design variables.

4.2 Gaussian Process (GP) Metamodeling for Vehicle Dynamical Responses

Since the functions $f_1(\boldsymbol{x}_0)$ through $f_5(\boldsymbol{x}_0)$ are expensive to evaluate via the MotionView® vehicle simulation model, metamodels are fitted to replace them in the optimization process. The enhanced GP modeling method introduced in Section 2.2 is applied to fit GP models for the five functions, with 460 DoE (Design of Experiment) sample points generated by a space-filling optimal Latin hypercube sampling algorithm [26] for 46 variables. The two surface plots of the fitted GP models shown in Figures 8 and 9 showcase relatively high and low nonlinearities of the functions. In plotting the two response surfaces, the design variables other than the two plotted ones are set to baseline values, i.e., 0.5 under normalized ranges.

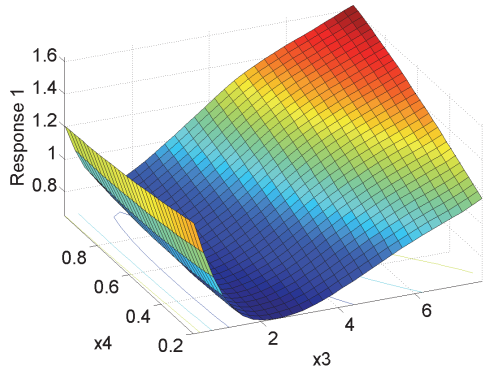

Figure 8. Response 1 (front left suspension tower vertical load) as a function of x_4 (front damper p_2 parameter) and x_3 (front damper C_1 parameter)

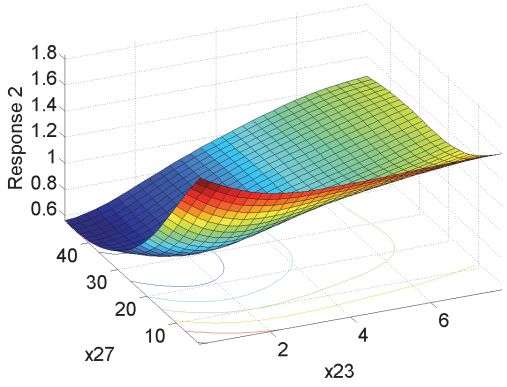

Figure 9. Response 2 (rear left suspension tower vertical load) as a function of x_{27} (rear bump stopper clearance) and x_{23} (rear damper C_1 parameter)

The advantage of using the enhanced GP modeling method is justified in two aspects. First, we show that the traditional optimization method without the two-stage parameter estimation technique (the key part of the enhanced method) has much lower efficiency in finding the globally optimal hyperparamters $\boldsymbol{\omega}$ of GP models than the enhanced method. As introduced in Section 2.1, the optimal $\boldsymbol{\omega}$ minmizes the objective function L (Eqn. (8)), and multiple optimization runs starting from different initial points have to be carried out to search for the global optimum. We compare the efficiency of the traditional and the enhanced GP modeling methods by using them to fit GP models for $f_1(\boldsymbol{x}_0)$ and recording the smallest objective function values achieved within 12 optimization runs. The default search range for each ω_i is $[-5, 5]$ for both methods, but the enhanced method carries out six preliminary optimization runs to refine the bounds and these six runs are counted in the total number. Sobol sequence [30] is used to generate space-filling initial points for both methods, and the whole test is repeated 10 times to assess the statistical performance. The results are plotted in Figure 10, where the smallest objective function values achieved by the enhanced method are consistently much smaller (better) than those by the traditional method.

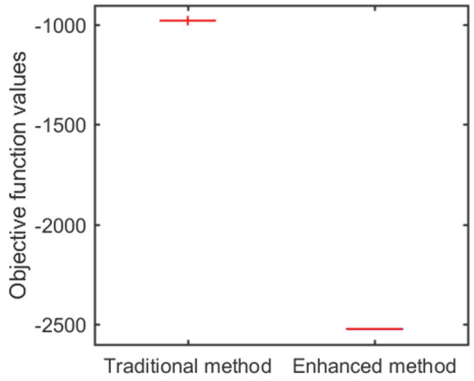

Figure 10. Comparison of the achieved smallest objective function values within 12 optimization runs using the traditional and the enhanced GP modeling methods

Second, to justify the use of GP models rather than other types of metamodels, a common metamodeling method, polynomial regression, is also tested to fit metamodels to the same datasets. As listed in Table 4, R-squared values based on 10-fold cross-validation (CV) are used to assess the accuracy of the two types of models. According to the R-squared values, GP models have overall higher accuracy than the polynomial models although the former has slightly lower accuracy than the latter for $f_4(\boldsymbol{x}_0)$ and $f_5(\boldsymbol{x}_0)$. This indicates the true functions of $f_4(\boldsymbol{x}_0)$ and $f_5(\boldsymbol{x}_0)$ have close-to-polynomial shapes, but the overall high R-squared values of the GP models demonstrate their robustness against the variation of the true function forms. In addition, GP models can provide uncertainty information at prediction locations which is useful for further studies on the problem such as additional sampling, while polynomial ones cannot. Therefore, GP models are overall the better choice for the metamodels in this design problem.

Table 4. Accuracy comparison of polynomial and GP models by 10-fold cross-validation R-squared values

	$f_1(\boldsymbol{x}_0)$	$f_2(\boldsymbol{x}_0)$	$f_3(\boldsymbol{x}_0)$	$f_4(\boldsymbol{x}_0)$	$f_5(\boldsymbol{x}_0)$
Polynomial model	0.887	0.941	0.987	0.979	0.991
GP model	0.966	0.983	0.997	0.928	0.986

4.3 CO Formulations for the MDO problem

Based on the formulation of the MDO problem (21) and the multidisciplinary structure (Figure 7), the different versions of the CO method discussed in Section 3 are formulated for the MDO problem as shown in Figure 11 through Figure 14. There are several points that deserve notice in these formulations.

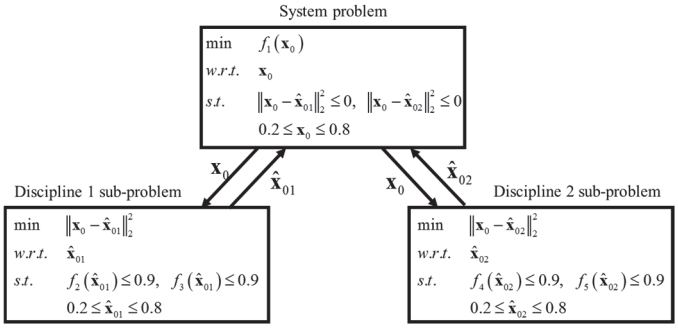

Figure 11. Original CO formulation for the MDO problem

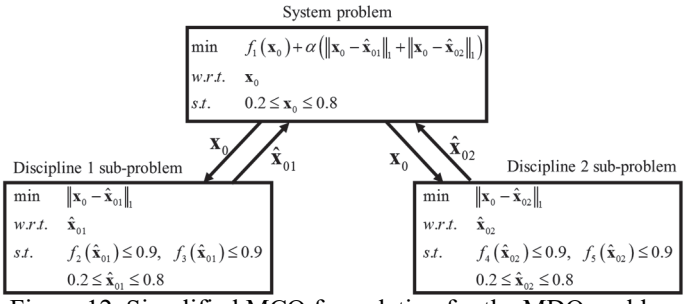

Figure 12. Simplified MCO formulation for the MDO problem ($\alpha = 0.01$)

1. For all the formulations, $\boldsymbol{x}_0$ is the 46 design variables shared by all the disciplines. $\hat{\boldsymbol{x}}_{01}$ and $\hat{\boldsymbol{x}}_{02}$ are copies of these variables in the two disciplinary problems. Analysis functions $f_1(\boldsymbol{x}_0)$ through $f_5(\boldsymbol{x}_0)$ are GP models fitted for the corresponding simulation models.

2. In addition to the original ECO (Figure 13), a simplified ECO formulation (Figure 14) is developed for comparison with the original one, as discussed at the end of Section 3.3.

3. In Figure 13, the response functions with tildes ($\tilde{f}_2(\boldsymbol{x}_0)$ through $\tilde{f}_5(\boldsymbol{x}_0)$) are linear models of nonlocal constraints corresponding to the $\tilde{\boldsymbol{c}}_j(\cdot)$ $(j \neq i)$ in Section 3.3. Our GP modeling code has a nice feature to predict both the mean response and the gradient for any input location; therefore, the linear models $\tilde{f}_2(\boldsymbol{x}_0)$ through $\tilde{f}_5(\boldsymbol{x}_0)$ can be easily built with the output of the GP models. If other metamodels are used which don't offer the gradient prediction, numerical methods such as finite differentiation may be used to build the linear models and no new metamodels need to be fitted.

4. The formulations of simplified MCO, original ECO and simplified ECO involve some penalty parameters. The chosen values for the parameters are noted in the figure captions.

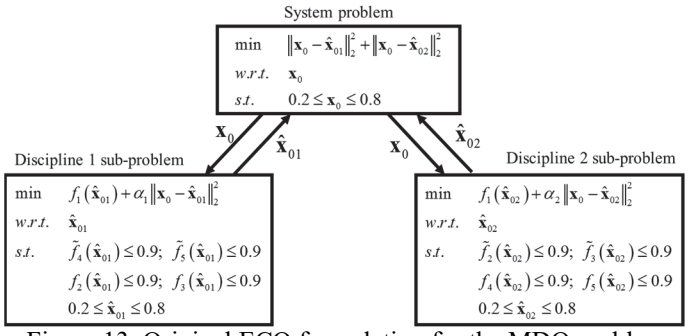

Figure 13. Original ECO formulation for the MDO problem ($\alpha_1 = \alpha_2 = 0.1$)

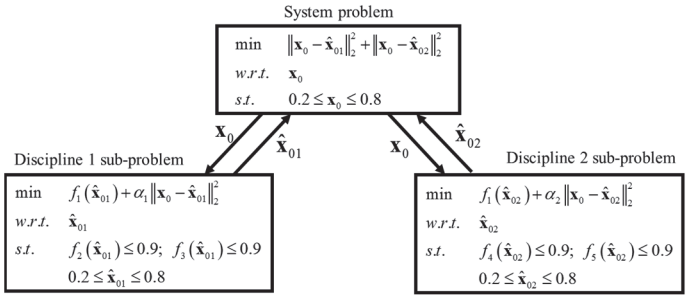

Figure 14. Simplified ECO formulation for the MDO problem ($\alpha_1 = \alpha_2 = 100$)

4.4 Optimization Results and Comparison

All the metamodel-based MDO problems with different versions of the CO method are solved in MATLAB® programming software. The all-at-once (AAO) method is also used to directly solve the MDO problem (21) and its solution serves as a benchmark for those of the CO methods [27]. The summary of optimization results is provided in Table 5. Among all different versions of the CO method studied, only the original ECO can obtain an optimal solution comparable with that obtained by the AAO method. The simplified ECO achieves a solution that reduces the global objective function moderately and roughly satisfies the 10% reduction constraints on $f_2(\boldsymbol{x}_0)$

Table 5. Optimization results of AAO and different versions of CO method

Method	$f_1(\boldsymbol{x}_0^*)$ (objective)	$f_2(\boldsymbol{x}_0^*)$	$f_3(\boldsymbol{x}_0^*)$	$f_4(\boldsymbol{x}_0^*)$	$f_5(\boldsymbol{x}_0^*)$	Convergence	Number of system-wide iterations
AAO	0.739	0.900	0.900	0.720	0.900	Yes	NA
Original CO	1.150	0.900	0.900	0.894	0.900	Yes	29
Simplified MCO	0.333	1.166	1.122	1.433	1.144	No	1000
Simplified ECO	0.823	0.900	0.901	0.776	0.901	No	1000
Original ECO	0.751	0.900	0.900	0.747	0.900	Yes	30

through $f_5(\boldsymbol{x}_0)$, but not all the constraints are strictly satisfied even after a large number (1000) of system-wide iterations. The remaining two methods, original CO and simplified MCO, either fail to effectively reduce the global objective function or cannot satisfy the constraints from all the disciplinary groups.

The original CO method converges within a small number of system-wide iterations with all the constraints satisfied, but the global objective function is increased rather than decreased compared with the baseline design. This is reasonable because in its formulation shown in Figure 11, the system problem imposes tight equality constraints on interdisciplinary variable consistency, allowing little feasible space where an optimum is searched for in the system problem.

In the simplified MCO method, a wide range of values for the penalty parameter α are tried, but none of them work well. The best performance of MCO is achieved when $\alpha = 0.01$. The drastic breakdown of the simplified MCO method is not of much surprise. Even though the interior-point optimization algorithm in MATLAB® is used to deal with the non-smoothness of the L1-norm consistency terms (Figure 12), the large number of design variables (listed in Table 3) shared by all disciplinary problems exacerbates the non-smoothness problem and may have neutralized this workaround. Therefore, the simplification of the original MCO method does not work for the MDO problem in this paper, and probably will not work for other similar engineering applications with large numbers of variables.

Of the most interest are the results of the simplified and the original ECO methods. Although it is obvious in Table 5 that the original ECO works much better than the simplified one in terms of both the quality of the final optimal solution and the number of system-wide iterations, the final solution obtained by the simplified ECO is very close to convergence and hence may be worthwhile. That is, only the responses $f_3(\boldsymbol{x}_0^*)$ and $f_5(\boldsymbol{x}_0^*)$ exceed the constraint target (0.9) by around 1% while the other constraints are satisfied, and the global objective function $f_1(\boldsymbol{x}_0^*)$ is reduced by a reasonable amount below the baseline value (1.0). In Section 3.3, it is shown that simplified ECO has an organizational advantage over the original ECO in that the former does not require interdisciplinary data exchange to linearize the nonlocal constraints. Considering this fact, a closer look at the detailed convergence behaviors of the two methods is taken to weigh the actual performance loss of the simplified ECO against the gain of its organizational advantage.

The convergence histories of the original and simplified ECO methods are shown in Figure 15 through Figure 17. It can be seen in Figure 15 that the conflict between reducing the global objective function $f_1(\boldsymbol{x}_0)$ and satisfying the constraints for $f_2(\boldsymbol{x}_0)$, $f_3(\boldsymbol{x}_0)$ and $f_5(\boldsymbol{x}_0)$ rapidly reaches a nearly feasible balance within about three iterations. Under this balance, the global objective can continue to decrease until all the convergence criteria are met. In contrast to this, the convergence behavior of the simplified ECO method has a different pattern shown in Figure 16 and Figure 17. Within the first 15 iterations, the same rough balance between the functions $f_1(\boldsymbol{x}_0)$, $f_2(\boldsymbol{x}_0)$, $f_3(\boldsymbol{x}_0)$ and $f_5(\boldsymbol{x}_0)$ as in Figure 15 is established, but with a much larger sacrifice of the $f_1(\boldsymbol{x}_0)$ value. Around this trade-off space, the optimization then proceeds along a path on which $f_1(\boldsymbol{x}_0)$ continuously decreases, but the same convergence criteria as in the original ECO case cannot be strictly satisfied even after 1000 iterations (Figure 17). Although slightly loosened convergence criteria can be met within the 1000 iterations, the achieved global objective value $f_1(\boldsymbol{x}_0^*)$ is still much worse than that of the original ECO.

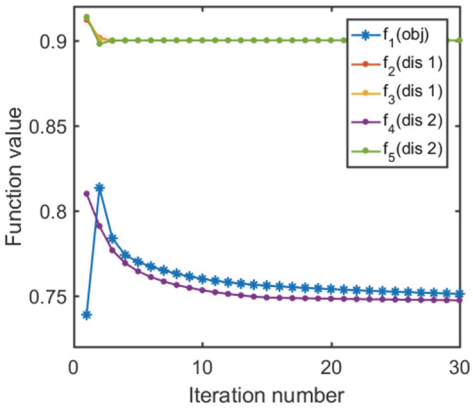

Figure 15. Convergence history of the original ECO method (convergent in 30 iterations)

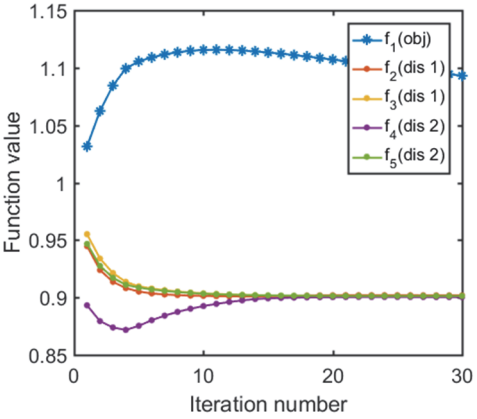

Figure 16. Convergence history of the simplified ECO method (the first 30 iterations, not yet convergent)

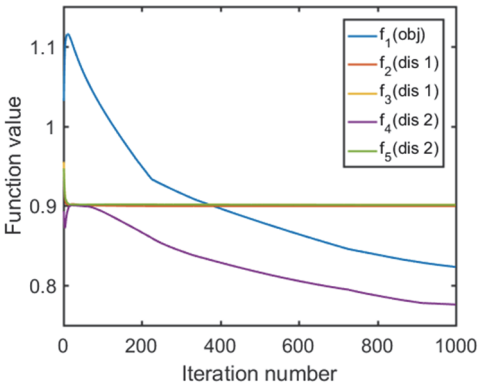

Figure 17. Convergence history of the simplified ECO method. The convergence is not achieved within the allowed number (1000) of iterations, but the final iterate is almost convergent.

Table 6. Verification of response function values at two optimal solutions

		$f_1(\boldsymbol{x}_0^*)$ (objective)	$f_2(\boldsymbol{x}_0^*)$	$f_3(\boldsymbol{x}_0^*)$	$f_4(\boldsymbol{x}_0^*)$	$f_5(\boldsymbol{x}_0^*)$
Solution 1 (AAO)	Metamodel responses	0.739	0.900	0.900	0.720	0.900
	Verification result	0.780	0.894	0.905	0.830	0.914
Solution 2 (ECO)	Metamodel responses	0.751	0.900	0.900	0.747	0.900
	Verification result	0.783	0.897	0.905	0.824	0.913

Based on the above observations, a clearer distinction of optimization performance between the original and the simplified ECO methods is identified, the mechanism of which can be explained by the difference between their formulations. The only difference between both formulations (Figure 13 and Figure 14) is the existence or nonexistence of the linear models of nonlocal constraints. In the original ECO method, due to the linear constraints, every disciplinary sub-problem has more "knowledge" about the constraint condition in the other discipline. Although the "knowledge" is limited to only the linear term in the context of Taylor expansion series, it is sufficient to efficiently guide the local solutions towards globally feasible design solutions, as is observed in Figure 15. Without the linearized constraints as in the simplified ECO method, the only way for the disciplinary sub-problems to "know" the constraint condition in the other discipline is via the penalty term in the sub-problem objective functions. However, the information passed by this penalty term is much inefficient at communicating the constraint condition of the other disciplines; hence the global feasibility is less strictly achieved as shown in Figure 17, where the same convergence criteria as used in the original ECO formulation cannot be met within 1000 iterations. Also, a much larger penalty parameter (100 as opposed to 0.1 in original ECO) has to be used to ensure sufficient global consistency and feasibility of the solution. This results in the expense of having to explore a much smaller feasible space and consequently a low-quality solution compared with that obtained by the original ECO method.

To sum up, among all different versions of the CO method tested, the simplified and the original ECO methods manifest noticeable performance advantages in solving the vehicle suspension parameter design problem. The choice between the two methods is a trade-off between ease of application and convergence performance. While the original ECO method can efficiently converge to a high-quality optimal solution, the linear models of nonlocal constraints required in each of the disciplinary sub-problems incur a high amount of interdisciplinary data exchange and extra function evaluations, which is undesirable or even prohibitive for applications to large-scale MDO problems. The simplified ECO method doesn't incorporate the linear models of nonlocal constraints, and therefore achieves complete independence between different discipline groups. This enables the parallelization of solving disciplinary sub-problems and is especially advantageous for large-scale MDO problems. The downside of the simplified ECO method, however, is worse convergence behavior and solution quality than that of the original ECO method.

4.5 Verification of the Optimal Solutions

To verify the accuracy of the GP models used to replace the expensive simulation models in this study, two optimal solutions obtained by the AAO and the original ECO methods are verified by calculating the true simulation responses at the corresponding optimal design points. The verification results are listed in Table 6. In each of the two cases, the differences between the GP model prediction values and the true response values are mostly small, with the maximum error percentage being around 10% for response function $f_4(\boldsymbol{x}_0)$. These results confirm the sufficient accuracy of the GP models used in this design problem.

5 CONCLUSIONS

Modern vehicle suspension design problems are faced with two major challenges: their large number of design variables and the proper choice of an MDO architecture. In this paper, an enhanced Gaussian process (GP) modeling method is developed and several versions of the collaborative optimization (CO) method are surveyed to address the two challenges. A vehicle suspension parameter design problem is solved to test the proposed GP modeling method and different versions of the CO method. The GP models are fitted efficiently and their high accuracy is verified. Among all the versions of the CO method, the most promising ones are the original and the simplified ECO methods, the pros and cons of which are a trade-off between ease of implementation and convergence performance. The original ECO method manifests superior convergence performance by efficiently achieving an optimal solution comparable to the one obtained by the all-at-once (AAO) method. The key contributors to such performance are the linear approximations of nonlocal constraints. However, these approximations also bring about expensive extra function evaluations and interdisciplinary data communication, which impairs the parallelization of solving disciplinary sub-problems and may become prohibitive for large-scale MDO applications. In contrast, the simplified ECO method without the linearly approximated constraints does not come with these organizational downsides, but as a compromise, its convergence performance decreases and large penalty parameters must be applied to ensure a desired level of global consistency of the design solutions.

This study produces two valuable outcomes for future vehicle suspension design problems and other engineering problems that face high-dimensionality and MDO challenges. One is that the proposed enhanced GP modeling method can be readily applied to any high-dimensional (e.g. more than 40) GP modeling task to save the time and computation cost while maintaining the desired level of model accuracy. The other is that insights drawn from the comparison between different versions

of the CO method provide guidance for future industrial applications of these methods. The original ECO method has the best optimization performance of all the versions of the CO method, and the simplified ECO method can be an alternative for practitioners who wish to trade a certain level of optimization performance for ease of implementation and independence of disciplinary analyses.

ACKNOWLEDGMENTS

Grant supports from National Science Foundation (CMMI-1537641) and Toyota Motor North America, Inc. are greatly appreciated.

REFERENCES

1. IM, K., *Development and application of the collaborative optimization architecture in a multidisciplinary design environment.* 1995.
2. DeMiguel, A.-V. and W. Murray. *An analysis of collaborative optimization methods*. in *8th Symposium on Multidisciplinary Analysis and Optimization*. 2000.
3. De Miguel, A.-V., *Two decomposition algorithms for nonconvex optimization problems with global variables*. 2001, Stanford University.
4. Roth, B. and I. Kroo. *Enhanced collaborative optimization: application to an analytic test problem and aircraft design*. in *12th AIAA/ISSMO multidisciplinary analysis and optimization conference*. 2008.
5. Roth, B.D., *Aircraft family design using enhanced collaborative optimization*. 2008: ProQuest.
6. Kennedy, M.C. and A. O'Hagan, *Predicting the output from a complex computer code when fast approximations are available.* Biometrika, 2000. **87**(1): p. 1-13.
7. Martin, J.D. and T.W. Simpson, *Use of kriging models to approximate deterministic computer models.* AIAA journal, 2005. **43**(4): p. 853-863.
8. Plumlee, M. and D.W. Apley, *Lifted Brownian kriging models.* Technometrics, 2016(just-accepted).
9. Sacks, J., et al., *Design and analysis of computer experiments.* Statistical science, 1989: p. 409-423.
10. Simpson, T.W., et al., *Metamodels for computer-based engineering design: survey and recommendations.* Engineering with computers, 2001. **17**(2): p. 129-150.
11. Xu, H., et al., *A descriptor-based design methodology for developing heterogeneous microstructural materials system.* Journal of Mechanical Design, 2014. **136**(5): p. 051007.
12. Kennedy, M.C. and A. O'Hagan, *Bayesian calibration of computer models.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2001. **63**(3): p. 425-464.
13. Bayarri, M.J., et al., *A framework for validation of computer models.* Technometrics, 2007. **49**(2): p. 138-154.
14. Arendt, P.D., D.W. Apley, and W. Chen, *Quantification of model uncertainty: Calibration, model discrepancy, and identifiability.* Journal of Mechanical Design, 2012. **134**(10): p. 100908.
15. Arendt, P.D., et al., *Improving identifiability in model calibration using multiple responses.* Journal of Mechanical Design, 2012. **134**(10): p. 100909.
16. Arendt, P.D., D.W. Apley, and W. Chen, *A preposterior analysis to predict identifiability in the experimental calibration of computer models.* IIE Transactions, 2016. **48**(1): p. 75-88.
17. Rideout, D.G., P.Y. Papalambros, and J.L. Stein, *Analytical target cascading in automotive vehicle design.* Ann Arbor, 2003. **1001**: p. 48109.
18. Kim, H.M., W. Chen, and M.M. Wiecek, *Lagrangian coordination for enhancing the convergence of analytical target cascading.* AIAA journal, 2006. **44**(10): p. 2197-2207.
19. Kang, D., et al., *Robust design optimization of suspension system by using target cascading method.* International Journal of Automotive Technology, 2012. **13**(1): p. 109-122.
20. Kang, N., M. Kokkolaras, and P.Y. Papalambros, *Solving multiobjective optimization problems using quasi-separable MDO formulations and analytical target cascading.* Structural and Multidisciplinary Optimization, 2014. **50**(5): p. 849-859.
21. Cressie, N., *The origins of kriging.* Mathematical geology, 1990. **22**(3): p. 239-252.
22. Rasmussen, C.E., *Gaussian processes for machine learning.* 2006.
23. Roustant, O., D. Ginsbourger, and Y. Deville, *DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization.* 2012.
24. MacDonald, B., P. Ranjan, and H. Chipman, *Gpfit: an r package for fitting a Gaussian process model to deterministic simulator outputs.* Journal of Statistical Software, 2015. **64**(i12).
25. Ranjan, P., R. Haynes, and R. Karsten, *A computationally stable approach to Gaussian process interpolation of deterministic computer simulation data.* Technometrics, 2011. **53**(4): p. 366-378.
26. Jin, R., W. Chen, and A. Sudjianto, *An efficient algorithm for constructing optimal design of computer experiments.* Journal of Statistical Planning and Inference, 2005. **134**(1): p. 268-287.
27. Martins, J.R. and A.B. Lambe, *Multidisciplinary design optimization: a survey of architectures.* AIAA journal, 2013. **51**(9): p. 2049-2075.
28. Alexandrov, N.M. and R.M. Lewis, *Analytical and computational aspects of collaborative optimization for multidisciplinary design.* AIAA journal, 2002. **40**(2): p. 301-309.
29. Ryberg, A.-B., R. Domeij Bäckryd, and L. Nilsson, *Metamodel-based multidisciplinary design optimization for automotive applications*. 2012: Linköping University Electronic Press.
30. Sobol', I.y.M., *On the distribution of points in a cube and the approximate evaluation of integrals.* Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, 1967. **7**(4): p. 784-802.