Copyright © 2010 Tech Science Press CMES, vol.66, no.3, pp.187-213, 2010

Sequential Approximate Optimization Procedure based on Sample-reusable Moving Least Squares Meta-model and its Application to Design Optimizations

Jin Yeon Cho[1] **and Min Hwan Oh**[1]

Abstract: In this work, a sample-reusable sequential approximate optimization (SAO) procedure is suggested. The suggested sequential approximate optimization procedure utilizes a newly proposed sample-reusable meta-model along with the trust region algorithm. Domain of design is sequentially updated to search for the optimal solution through the trust region algorithm, and the system response in the updated design region at each sequential stage is approximated by the proposed sample-reusable meta-model. The proposed sample-reusable meta-model is based on the moving least squares(MLS) approximation scheme. Thanks to the merits of moving least squares scheme, the proposed meta-model can fully utilize the previously sampled responses as well as the currently sampled responses of the system, and consequently it makes it possible to enhance the accuracy and robustness of meta-model (often called response surface) for system response.
Through the typical optimization problems, the performance of proposed approach is investigated. After the investigations, a preliminary optimal design of compound helicopter is carried out by using the proposed sample-reusable sequential approximate optimization procedure as a practical example of design optimization.

Keywords: sequential approximate optimization(SAO), sample-reusable meta-model, moving least squares(MLS), trust region algorithm, response surface.

1 Introduction

Meta-model based design optimization (MBDO) has been one of the major trends in design optimizations of complex engineering systems in the past two decades [Wang, Shan (2007)]. The major reason to use meta-model (such as polynomial based regression, Kriging, thin plate splines, radial basis function, moving least squares, etc. [Lancaster, Salkauskas (1981); Atluri, Cho, Kim (1999); Jones (2001); Myers and Montgomery (2002); Ho, Yang, Ni, Wong (2007); McDonald, Grantham,

[1] Department of Aerospace Engineering, Inha University, Incheon 402-751, Republic of Korea.

Tabor, Murphy (2007); Panda and Manohar (2008)]) instead of high-fidelity model is the expensive computational cost which is required to obtain the response of high-fidelity model. Of course, computational cost per arithmetic operation has been greatly reduced compared to 20 years ago, because of the startling progress in computing hardware technologies. On the contrary, the fidelity of model in engineering practice has been enormously increased according to the affordability of computing hardware and ever-increasing requirements for more reliable, more efficient, and more competitive design. Therefore, even nowadays it is still common that it takes couple of days to carry out high-fidelity simulation of complex systems such as aerospace vehicle, automobiles, and others.

Because of the aforementioned research driving force, a considerable research effort has been given to meta-model based design optimization (MBDO). As a result, several optimization strategies have been proposed in the context of meta-model based design optimization [Schmit, Jr. and Farshi (1974); Sacks, Welch, Mitchell, Wynn (1989); Jones, Schonlau, Welch (1998); Rodriguez, Renaud, Watson (1998); Alexandrov, Lewis, Gumbert, Green, Newman (2001); Myers and Montgomery (2002)]. Among these, successive approximate optimization(SAO) procedure is one of the mostly preferred strategies in engineering practice because of its computational efficiency and guaranteed convergence to local optima [Fletcher (1987); Rodriguez, Renaud, Watson (1998); Alexandrov, Lewis, Gumbert, Green, Newman (2001)].

In successive approximate optimization procedure, a repetitive strategy is adopted. In each sequential step, region of design space is restricted to local region, and the sub-optimization problem is considered in that region. The sub-optimization problem in the restricted region is carried out along with meta-model according to the following procedure. At first, appropriate locations in the restricted design space are selected through so called DoE(design of experiments) techniques, and system responses are obtained by the simulation of high-fidelity model at the selected locations. The procedure is often called sampling. After sampling, the sampled responses are utilized to construct a meta-model (usually quadratic polynomial regression model [Sobieski, Manning, Kroo (1998)]) which is a surrogate of the system response of high-fidelity model. With the constructed meta-model, sub-optimization is carried out within the restricted design space to select a probable optimal solution. If the obtained probable optimal solution is not the converged solution, then the next sequential restricted region is updated. And the same procedure is repeated until the probable optimal solution of sub-problem converges to the optimal solution.

In sequential approximate optimization(SAO), meta-model is constructed by using new sampling points in each sequential step with no regard to the previously sampled points, although the previous sampling points may provide valuable information about the system response. Taking into account the expensive computational cost paid for sampling responses, recycling the pre-sampled responses may be a smart strategy to save a computational cost.

In the line of thought, a novel sequential approximate optimization(SAO) procedure, based on sample-reusable meta-model, is proposed in this work. To construct the sample-reusable meta-model, moving least squares method is employed instead of the conventional polynomial regression meta-model because of its favorable characteristics [Atluri (2004), Atluri (2005)]. In numerical examples, the performance of proposed method is investigated through the representative optimization problems including Branin function, Hosaki function, and Haupt function. As a practical example of design optimization, a preliminary design optimization of compound helicopter is performed by using the proposed sample-reusable sequential approximate optimization(SAO) procedure, and its potential is observed in practical point of view.

2 Sequential Approximate Optimization based on Sample-reusable MLS Meta-model

As mentioned in introduction, there are three main parts in the sequential approximate optimization procedure. The first one is how to restrict the region of design in each sequential step, the second one is how to select the sampling position of response in the restricted region, and the third one is how to construct an accurate surrogate model which could reflect the real response of the system appropriately. In forthcoming subsections, the three main parts of SAO procedure used in this work are described.

2.1 Trust Region Algorithm for Restriction of Region of Design

Trust region algorithm was originally proposed in the context of Newton's method, and successfully applied to design optimization problems [Sorensen (1982); Fletcher (1987); Rodriguez, Renaud, Watson (1998)]. Trust region algorithm restricts the step size first and then determines the probable optimal solution in the restricted trust region, whereas conventional optimization algorithms determine the search direction first and then determine the step size. The trust region algorithm can be summarized as follows.

Consider a design optimization problem, where $\mathbf{x}$ is the design variable, $f(\mathbf{x})$ is the objective function, and Ω is the region of design. Without loss of generality, one may assume that $\mathbf{x}^{(k)}$ is the probable optimum point at the current (k)-*th* step, and $\tilde{f}^{(k)}(\mathbf{x})$ is an approximated meta-model of objective function $f(\mathbf{x})$ over the restricted trust region $\Omega^{(k)}$ with radius $\Delta^{(k)}$ at the (k)-*th* step as shown in Fig. 1.

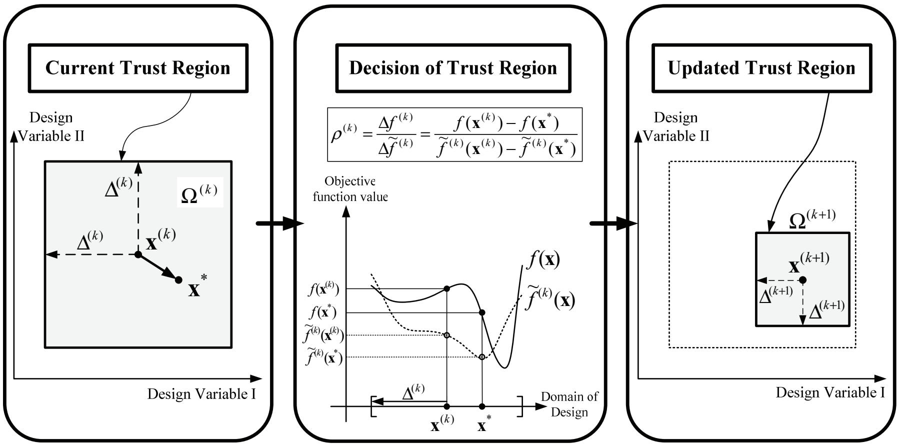


Figure 1: Sketch of the concept of trust region algorithm.

Then at first the trust region algorithm seeks the solution $\mathbf{x}^*$ of the resulting sub-problem for the next $(k+1)$-*th* sequential step as denoted in Eq. 1.

$$\mathbf{x}^* = ARG\left[\min_{\mathbf{x}\in\Omega^{(k)}} \tilde{f}^{(k)}(\mathbf{x})\right] \tag{1}$$

where $\Omega^{(k)} = \left\{\mathbf{x} : \left\|\mathbf{x} - \mathbf{x}^{(k)}\right\| \leq \Delta^{(k)}\right\} \cap \Omega$

And the trust ratio $\rho^{(k)}$ between the actual reduction $\Delta f^{(k)} = f(\mathbf{x}^{(k)}) - f(\mathbf{x}^*)$ and predicted reduction $\Delta\tilde{f}^{(k)} = \tilde{f}^{(k)}(\mathbf{x}^{(k)}) - \tilde{f}^{(k)}(\mathbf{x}^*)$ is evaluated as shown in Eq. 2.

$$\rho^{(k)} = \frac{\Delta f^{(k)}}{\Delta\tilde{f}^{(k)}} = \frac{f(\mathbf{x}^{(k)}) - f(\mathbf{x}^*)}{\tilde{f}^{(k)}(\mathbf{x}^{(k)}) - \tilde{f}^{(k)}(\mathbf{x}^*)} \tag{2}$$

To update the next probable optimum point and the size of trust region, the following decision algorithm is utilized.

[Decision Algorithm I]

$$\begin{cases} \text{If } \rho^{(k)} < \kappa_1, \text{ then } \Delta^{(k+1)} = c_1\left\|\mathbf{x}^* - \mathbf{x}^{(k)}\right\| \\ \text{If } \rho^{(k)} > \kappa_2 \text{ and } \left\|\mathbf{x}^* - \mathbf{x}^{(k)}\right\| = \Delta^{(k)}\text{ , then } \Delta^{(k+1)} = c_2\Delta^{(k)} \\ \text{Otherwise, } \Delta^{(k+1)} = \Delta^{(k)} \end{cases} \tag{3}$$

$$\begin{cases} \text{If } \rho^{(k)} \leq 0, \text{ then } \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}, \\ \text{else } \mathbf{x}^{(k+1)} = \mathbf{x}^* \end{cases}$$

And the above is repeated until the probable optimal point $\mathbf{x}^{(k+1)}$ converges to the optimal solution. In decision algorithm I (Eq. 3), κ_1, κ_2, c_1, and c_2 denote the algorithmic constants. Customarily the constants are adopted as the values suggested by Fletcher (1987) ($\kappa_1 = 0.25$, $\kappa_2 = 0.75$, $c_1 = 0.25$, and $c_2 = 2$).

However, it is noted that the constants could be modified according to the problem of interest [Alexandrov, Lewis, Gumbert, Green, Newman (2001)]. Furthermore, the size of next trust region can be also controlled by the size $\Delta^{(k)}$ of current trust region instead of the distance $\left\|\mathbf{x}^* - \mathbf{x}^{(k)}\right\|$ as shown in Eq. 4 [Rodriguez, Renaud, Watson (1998)].

[Decision Algorithm II]

$$\begin{cases} \text{If } \rho^{(k)} < \kappa_1, \text{ then } \Delta^{(k+1)} = c_1\Delta^{(k)} \\ \text{If } \rho^{(k)} > \kappa_2 \text{ and } \left\|\mathbf{x}^* - \mathbf{x}^{(k)}\right\| = \Delta^{(k)}, \text{ then } \Delta^{(k+1)} = c_2\Delta^{(k)} \\ \text{Otherwise, } \Delta^{(k+1)} = \Delta^{(k)} \end{cases} \qquad (4)$$

$$\begin{cases} \text{If } \rho^{(k)} \le 0, \text{ then } \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}, \\ \text{else } \mathbf{x}^{(k+1)} = \mathbf{x}^* \end{cases}$$

It is noted that the algorithm I (Eq. 3) can reduce the trust region size faster than the algorithm II (Eq. 4). Because of the feature, the decision algorithm I usually shows faster convergence than the decision algorithm II, once it starts to converge to the optimal point. On the contrary, the algorithm I is more susceptible to converging to local optimal point compared with the algorithm II. Additionally, it is noted that size of trust region is maintained or increased in both algorithms when trust ratio $\rho^{(k)}$ is larger than κ_2, even though trust ratio, larger than 1, implies that the metamodel does not describe the system response appropriately.

Based on the observations, the following modified decision algorithm is also considered to inherit the merits of both algorithms as well as alleviate the drawbacks of algorithms I and II in this work.

[Decision Algorithm III]

$$\begin{cases} \text{If } \rho^{(k)} < \kappa_1 \text{ or } \rho^{(k)} > \kappa_3, \text{ then } \Delta^{(k+1)} = \min\left(c_1\Delta^{(k)},\ c_3\left\|\mathbf{x}^* - \mathbf{x}^{(k)}\right\|\right) \\ \text{If } \kappa_2 < \rho^{(k)} < \kappa_3 \text{ and } \left\|\mathbf{x}^* - \mathbf{x}^{(k)}\right\| = \Delta^{(k)}, \text{ then } \Delta^{(k+1)} = c_2\Delta^{(k)} \\ \text{Otherwise, } \Delta^{(k+1)} = \Delta^{(k)} \end{cases} \qquad (5)$$

$$\begin{cases} \text{If } \rho^{(k)} \le 0, \text{ then } \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}, \\ \text{else } \mathbf{x}^{(k+1)} = \mathbf{x}^* \end{cases}$$

In algorithm III (Eq. 5) algorithmic constants κ_1, κ_2, κ_3, c_1, c_2, and c_3 are chosen as $\kappa_1 = 0.25$, $\kappa_2 = 0.75$, $\kappa_3 = 4$, $c_1 = 0.25$, $c_2 = 2$ and $c_3 = 10$.

2.2 Design of Experiments for Selection of Sample Points

Selection of sample points, which is referred to as DoE(design of experiments), is important to construct a meta-model which reflects the behavior of considered system appropriately. Due to the reason, various sampling schemes have been proposed. Representative sampling schemes are FD (factorial design), CCD (central composite design), FCD (face centered design), BBD (Box-Behnken Design), LHS (Latin hypercube sampling), and others [McKay, Beckman, Conover (1979); Myers and Montgomery (2002)].

Among them, LHS(Latin hybercube sampling) method [McKay, Beckman, Conover (1979); Park (1994); Zadeh, Toropov, Wood (2009)] is considered in this work because of its favorable characteristics. It makes it possible to choose the number of sampling points independent of the number of design variables. Further, Latin hypercube sampling points could be arranged to have a good space filling property [Johnson, Moore, Ylvisaker (1990); Tang (1993); Ye (1998)]. LHS selects only one sample point in each level of every design variables as shown in Fig. 2. Therefore one can obtain the quadratic regression model only with 10 sampling points for the problems with 3 design variables. To construct the quadratic regression surface for the problems with 3 design variables, factorial sampling design requires 27 sampling points, and central composite design requires 15 points at least.

In this work, lattice-based LHS algorithm [Hwang (2000)] is utilized to generate the Latin hypercube sampling points. The utilized lattice-based LHS algorithm is simple to implement and may generate sampling points similar to GLP(good lattice point) algorithm, while it preserves the intrinsic nature of Latin hypercube sampling [Fang, Wang (1994); Fang, Wang, Bentler (1994); Hwang (2000)]. Algorithm for lattice-based LHS is summarized as shown in box.

It is noted that $\mathrm{mod}(a,b)$ denotes the remainder of division of a by b. In implementation Q_{d1} can be chosen randomly or intentionally. It is recommended to take a value Q_{d1} for a given number of sampling points such that the sampling points have a good space filling property. In mapping of the sampling level matrix $\mathbf{Q}$ into design space, mid-point or random point in the corresponding level could be utilized as a sampling point as shown in Fig. 2. The typical distribution of generated sampling points in two dimensional design space is presented in Fig. 3.

2.3 Moving Least Squares Approximation Scheme

Moving least squares approximation scheme has been widely used in the field of computational mechanics because of its flexibility in dealing with irregular distribution of sampling points and in achieving the requirement for smoothness of the approximated function [Atluri, Han, Shen (2003)]. Further, information from

[Lattice-based LHS Algorithm]

Define the number of design variables D.
Choose the number of sampling N such that $N > D$.
Set $[Q_{11}, Q_{12}, \cdots, Q_{1N}] = [1, 2, \cdots, N]$

For ($2 \le d \le D$)
  If (N is prime number) {take integer Q_{d1} such that $2 \le Q_{d1} \le N-1$}
  elseif ($(N+1)$ is prime number) {take integer Q_{d1} such that $2 \le Q_{d1} \le N$}
  else {take integer Q_{d1} such that $2 \le Q_{d1} \le N-1$
    and Q_{d1} is not common divisor of N
    and Q_{d1} is not a multiple of common divisor of N}
end

For ($2 \le k \le N,\ 2 \le d \le D$)
  If (N is prime number){
    If ($\mathrm{mod}(kQ_{d1}, N) = 0$) { $Q_{dk} = N$}
    else { $Q_{dk} = \mathrm{mod}(kQ_{d1}, N)$}
  }
  elseif ($(N+1)$ is prime number) { $Q_{dk} = \mathrm{mod}(kQ_{d1}, N+1)$}
  else{
    If ($\mathrm{mod}(kQ_{d1}, N) = 0$) { $Q_{dk} = N$}
    else { $Q_{dk} = \mathrm{mod}(kQ_{d1}, N)$}
  }
end

Construct the sampling level matrix $\mathbf{Q}$ such that

$$\mathbf{Q} = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} & \cdots & Q_{1N} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ Q_{D1} & Q_{D2} & Q_{D3} & \cdots & Q_{DN} \end{bmatrix}$$

Each column of $\mathbf{Q}$ corresponds to each sampling point.
(where Q_{dk} deontes the level of d-th design variable of k-th sampling point.)

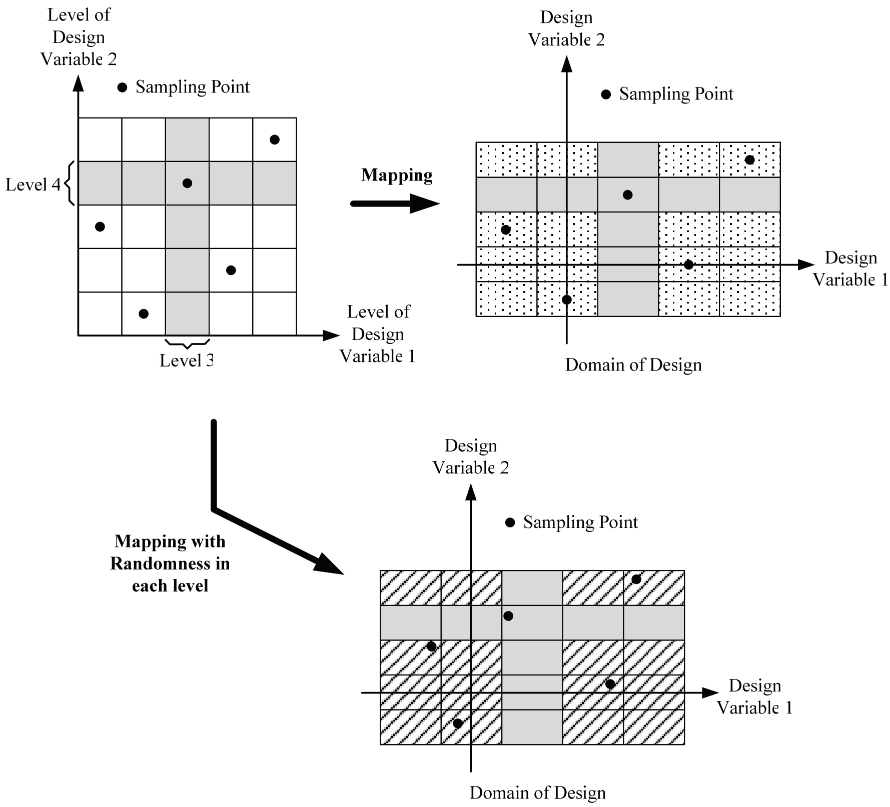


Figure 2: Concept of Latin hypercube sampling.

the additional sampling points could be easily considered with no difficulty [Atluri (2004)].

Due to the reasons, the moving least squares approximation scheme may be a good candidate for meta-model rather than the conventional polynomial regression model in sequential approximate optimization procedure. Therefore, a meta-model, which is based on the moving least squares approximation scheme, is considered for sequential approximate optimization in this work. In this section, the moving least squares approximation scheme is briefly reviewed.

Consider a continuous function $f(\mathbf{x})$ defined on a domain of design Ω, where the design variable $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_D \end{bmatrix}^T$ is a $D-$dimension vector. And assume that the function values at the sampling points $\mathbf{x}_I$ $(1 \leq I \leq N)$ in Ω are given as $f(\mathbf{x}_I)$ $(1 \leq I \leq N)$. Then in moving least squares scheme, the following form $\tilde{f}(\mathbf{x})$

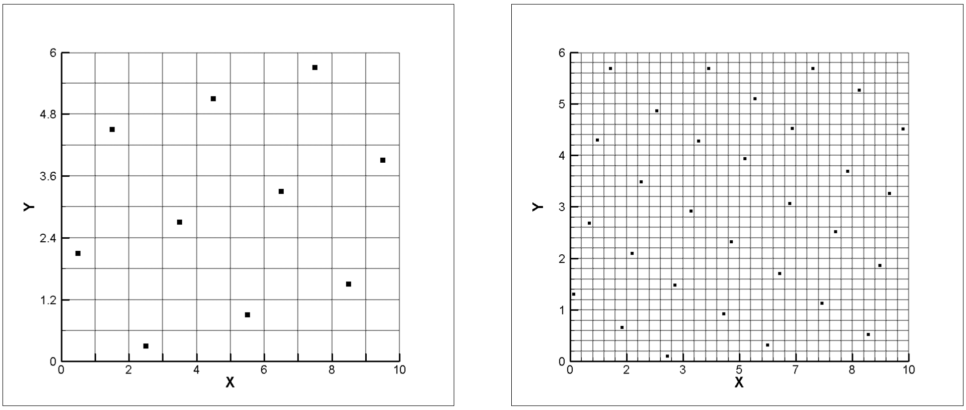


Figure 3: Typical distribution of Latin hypercube sampling.

is defined to approximate the function $f(\mathbf{x})$.

$$\forall\ \mathbf{x} \in \Omega,\ \tilde{f}(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) = \sum_{i=1}^{m} p_i(\mathbf{x})a_i(\mathbf{x}) \tag{6}$$

where $\mathbf{p}(\mathbf{x})$ is a complete monomial basis of order m, and $\mathbf{a}(\mathbf{x})$ is a vector containing coefficients $a_i(\mathbf{x})$ $(1 \le i \le m)$. The basis $\mathbf{p}(\mathbf{x})$ is selected to contain constant '1', and to be linearly independent over some set of m among the given N points in Ω. In this work, quadratic monomial basis is selected. If the design variable $\mathbf{x}$ is a two dimension vector and its components are x and y, then the corresponding quadratic basis is written as

$$\mathbf{p}^T(\mathbf{x}) = \begin{bmatrix} 1 & x & y & x^2 & xy & y^2 \end{bmatrix} \tag{7}$$

The coefficient vector $\mathbf{a}(\mathbf{x})$ is determined by minimizing a weighted discrete L_2 error norm as follows.

$$\mathbf{a}(\mathbf{x}) = \underset{\mathbf{b}\in R^m}{ARG}\left\{[\mathbf{Pb}-\mathbf{F}]^T\,\mathbf{W}(\mathbf{x})\,[\mathbf{Pb}-\mathbf{F}]\right\} = \left[\mathbf{P}^T\mathbf{W}(\mathbf{x})\mathbf{P}\right]^{-1}\left[\mathbf{P}^T\mathbf{W}(\mathbf{x})\right]\mathbf{F} \tag{8}$$

where

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}^T(\mathbf{x}_1) \\ \mathbf{p}^T(\mathbf{x}_2) \\ \vdots \\ \mathbf{p}^T(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} p_1(\mathbf{x}_1) & \cdots & p_m(\mathbf{x}_1) \\ p_1(\mathbf{x}_2) & \cdots & p_m(\mathbf{x}_2) \\ & \vdots & \\ p_1(\mathbf{x}_N) & \cdots & p_m(\mathbf{x}_N) \end{bmatrix} \tag{9}$$

$$\mathbf{F} = \begin{Bmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \vdots \\ f(\mathbf{x}_N) \end{Bmatrix} \tag{10}$$

$$\mathbf{W}(\mathbf{x}) = \begin{bmatrix} W_1(\mathbf{x}) & 0 & \cdots & 0 \\ 0 & W_2(\mathbf{x}) & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & W_N(\mathbf{x}) \end{bmatrix} \tag{11}$$

The $N \times m$ matrix $\mathbf{P}$ consists of basis, and the vector $\mathbf{F}$ denotes the vector of given values $f(\mathbf{x}_I)$ at sample points $\mathbf{x}_I$ $(1 \le I \le N)$ in the domain Ω. The $N \times N$ diagonal matrix $\mathbf{W}(\mathbf{x})$ is composed of weight functions. The weight function $W_I(\mathbf{x})$ is associated with the sample point $\mathbf{x}_I$. The weight function $W_I(\mathbf{x})$ is selected to be non-negative for all $\mathbf{x}$, and the region of non-zero values is called the support. In computations, various kinds of weight functions can be adopted for MLS approximation procedure. The required condition for the continuity of the approximated function can be easily satisfied by changing the weight function in the MLS approximation. In this work, the following polynomial tensor-product weight function is used.

$$W_I(\mathbf{x}) = \prod_{\alpha=1}^{D} \left(1 - \left(\frac{|x_\alpha - (x_\alpha)_I|}{(R_\alpha)_I} \right)^2 \right)^4 \tag{12}$$

where x_α and $(x_\alpha)_I$ denote the α - th components of design variables $\mathbf{x}$ and $\mathbf{x}_I$, respectively, and $(R_\alpha)_I$ denotes the radius of support of weight function (radius of influence) for α - th component of design variable $\mathbf{x}_I$ in Ω.

2.4 Sample-reusable MLS(Moving Least Squares) Meta-model

As pointed out before, the main reason to use meta-model instead of real system response is the expensive computational cost for direct evaluation of system response. In case of optimization related to high-fidelity mathematical model, expensive computational cost should be paid to obtain the response even in one sampling location. Usually, the sampling cost overwhelms the computing cost to construct a meta-model. Therefore, it is essential to extract all the information from the costly responses of sampling points.

In most of sequential approximate optimization approaches, the sampling points at the previous steps are not considered, and only the responses of new sampling points in the current trust region are considered to construct the meta-model. However, if one closely looks at the sequential approximate procedure, one may easily notice that the previously sampled points may exist in the current trust region.

Based on the crucial investigation, we are aiming to fully utilize the information of previously sampled responses with no additional sampling cost.

If one uses the moving least squares scheme, one can improve the accuracy of meta-model by additional sampling points. We utilize this feature to enhance the quality of meta-model by using the previous sampling points.

In this work, we fully take advantage of the pre-sampled points not only in the current trust region but also near the current trust region under the nearness concept, since the behaviors of sampling points near the trust region may provide the valuable information about the system response.

The nearness is determined by the intersection of current trust region and the support of weight function associated with the sampling point. If the intersection is not empty, it is determined that the sampling point of interest is near the trust region, and its response is utilized to construct the moving least squares meta-model for the current trust region as shown in Fig. 4. By the procedure, the previously sampled points are fully utilized even when the points are not inside the current trust region. The procedure to construct the proposed sample-reusable MLS(moving least squares) meta-model can be written as follows.

[Procedure to obtain the sample-reusable MLS meta-model]

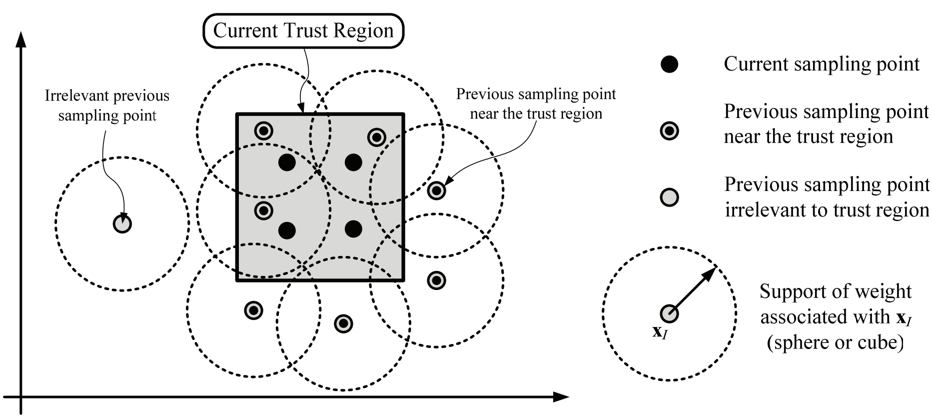


Figure 4: Concept of nearness for the trust region and previously sampled points.

(1) Assume that the total number of sampling points until last sequential step is TN, and the set of previously sampled points is given as $\mathscr{P} = \{\mathbf{x} : \mathbf{x} = \mathbf{x}_I^{(previous)},\ 1 \le I \le TN\}$.

(2) The current trust region $\Omega^{(k)}$ is given by the trust region algorithm.

(3) Choose the sampling point $\mathbf{x}_J^{(current)} (1 \le J \le N)$ in the current trust region $\Omega^{(k)}$ by lattice-based Latin hypercube method or others. (N denotes the number of sampling points at each step.)

(4) Construct the set of current sampling points as $\mathscr{C} = \{\mathbf{x} : \mathbf{x} = \mathbf{x}_J^{(current)},\ 1 \le J \le N\}$

(5) Obtain the system response at the current sampling point $\mathbf{x}_J^{(current)} (1 \le J \le N)$.

(6) Define the support size of weight for $\mathbf{x}_J^{(current)} (1 \le J \le N)$ to construct MLS meta-model.

(7) Adjust the support size of weight for the pre-sampled point $\mathbf{x}_I^{(previous)}$ $(1 \le I \le TN)$.

(8) Define the support set $S_I^{(previous)}$ of weight $W_I^{(previous)}(\mathbf{x})$ associated with $\mathbf{x}_I^{(previous)}$ as $S_I^{(previous)} = \left\{\mathbf{x} \in \Omega : W_I^{(previous)}(\mathbf{x}) \neq 0\right\}$

(9) Find the pre-sampled points near the trust region $\Omega^{(k)}$ by means of the nearness criterion.

$$\begin{cases} \mathbf{x}_I^{(previous)} \text{ is near the trust region } \Omega^{(k)} & \text{if } S_I^{(previous)} \cap \Omega^{(k)} \neq \phi \\ \mathbf{x}_I^{(previous)} \text{ is not near the trust region } \Omega^{(k)} & \text{if } S_I^{(previous)} \cap \Omega^{(k)} = \phi \end{cases}$$

(10) Make neighboring set $\mathscr{N}$ which consists of pre-sampled points near the trust region.

$$\mathscr{N} = \left\{\mathbf{x} : \mathbf{x} \in \mathscr{P} \text{ and } \mathbf{x} \text{ is near the trust region } \Omega^{(k)}\right\}$$

(11) Set $\mathscr{A} = \mathscr{C} \cup \mathscr{N}$.

(12) Construct a sample-reusable MLS meta-model with the sampling points in A by the moving least squares scheme with no additional sampling cost.

In this work, the radii of supports of weights for both of current and previous sampling points are selected as the same size of trust region for simplicity, even though those can be adopted arbitrarily if they do not induce singularity in moving least squares approximation.

In Fig. 5, flowchart for sequential approximate optimization procedure with sample-reusable MLS meta-model is presented. Optimization of sub-problem in trust region is carried out by using the genetic algorithm in this work.

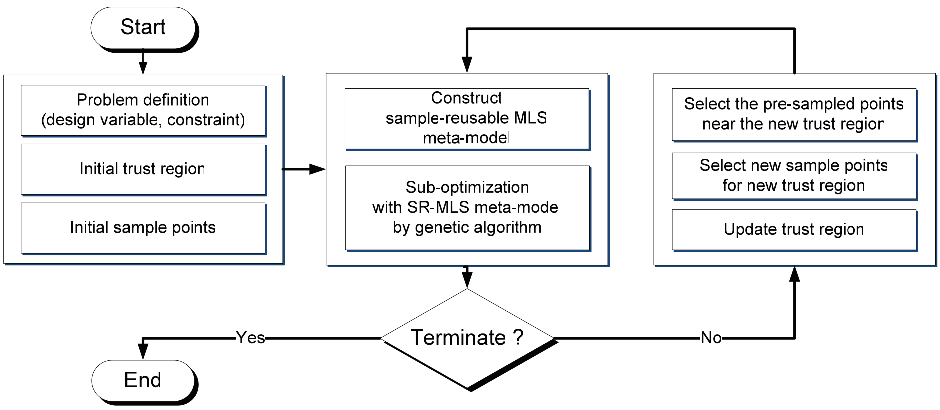
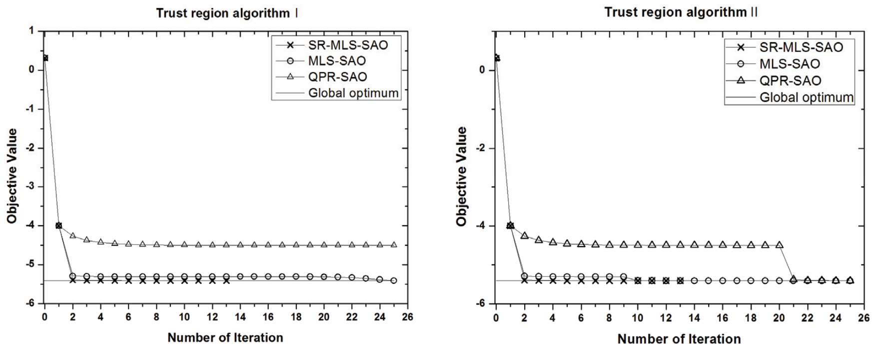


Figure 5: Flowchart for SAO procedure with sample-reusable meta-model.

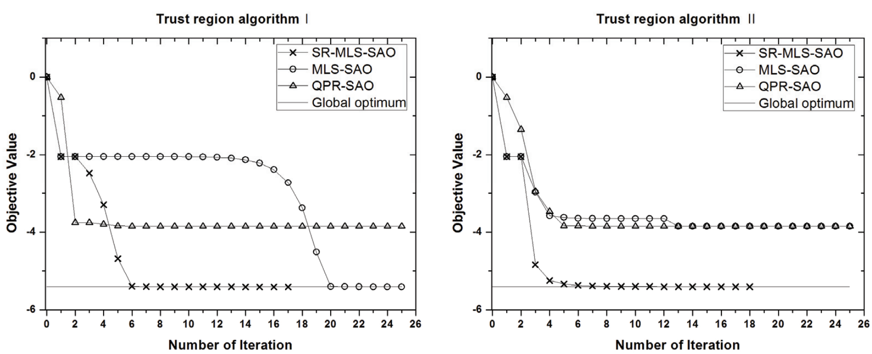
3 Numerical tests

For representative functions including Haupt, Hosaki, and Branin functions, optimizations are carried out by the proposed sample-reusable SAO procedure along with 10 points Latin hypercube sampling, and the performance is compared with those of conventional SAO procedure with quadratic polynomial regression meta-model and SAO procedure with MLS meta-model. In examples, in order to observe the tendency of convergence, the iteration is carried out until 25 steps, if the global solution is not found or if the rate of convergence is slow. In Figs. 6-11, sample-reusable SAO, SAO with MLS meta-model, and SAO with quadratic polynomial regression meta-model are denoted by ‘SR-MLS-SAO’, ‘MLS-SAO’, and ‘QPR-SAO’, respectively.



(a) (b)

Figure 6: Iteration history for optimization of Haupt function (1^{st} starting condition).



(a) (b)

Figure 7: Iteration history for optimization of Haupt function (2^{nd} starting condition).

3.1 Optimization of Haupt function

At first, Haupt function[Haupt, Haupt (1998)] is optimized by the proposed sample-reusable SAO procedure. Haupt function is highly nonlinear function and there are many local minima. Therefore, it is a very effective benchmarking function by which the performance of optimization procedure could be measured. Haupt function is written as follows.

$$f^{(Haupt)}(x,y) = x\sin(4x) + 1.1y\sin(2y) \tag{13}$$

Design space is chosen as $0 \leq x \leq 4$ and $0 \leq y \leq 4$, and the global optimum of Haupt function is -5.408 at location $(x,y) = (2.771,\ 2.456)$. In Fig. 6, the starting point is adopted as the center of design space, and the radius of initial trust region is chosen as 1/8 times the size of design space. Trust region algorithms I and II are utilized to obtain the results presented in Fig. 6a and Fig. 6b, respectively. In case of trust region algorithm I, result of Fig. 6a shows that the proposed SR-MLS-SAO(sample-reusable SAO) procedure finds the global optimum rapidly, whereas convergence of MLS-SAO is very slow, and QPR-SAO(quadratic polynomial regression meta-model SAO) converges to local minima and fails to find the global optimum. Similar to the case of trust region algorithm I in Fig. 6a, convergence rate of SR-MLS-SAO is faster than those of the others in case of trust region algorithm II as shown in Fig. 6b. From the results, it can be known that the proposed SR-MLS-SAO procedure is more reliable than the other SAO procedures.

In case of Fig. 7, the starting point is adopted as the lower-left corner of design space and the radius of initial trust region is selected as the size of entire design space. Generally, it is more difficult to find the global optimum with this starting condition compared with the starting condition used in Fig. 6, because the initial radius of trust region is too large to approximate the system behavior appropriately by meta-model as well as the starting point is far from the global optimum. Like the previous case of Fig. 6, trust region algorithms I and II are utilized to obtain the results presented in Fig. 7a and Fig. 7b, respectively.

In Fig. 7, the iteration path of SR-MLS-SAO seems to be similar to that of MLS-SAO during initial two steps. However, SR-MLS-SAO changes the iteration path after the second step, because the influence of previously sampled responses becomes larger as iteration goes on.

Similar to Fig. 6, the proposed sample-reusable SAO procedure gives the global optimal solution and shows rapid convergence even though it is difficult to find the global optimal solution with this given starting condition. From the results, it is confirmed that the proposed SR-MLS-SAO procedure is more robust compared with the other SAO procedures.

3.2 Optimization of Hosaki function

At second, Hosaki function [Bekey and Ung (1974)] is optimized by the proposed sample-reusable SAO procedure. Hosaki function is given by

$$f^{(Hosaki)}(x,y) = \left(1 - 8x + 7x^2 - \frac{7x^3}{3} + \frac{x^4}{4}\right) y^2 \exp(-y) \tag{14}$$

Design space is chosen as $0 \le x \le 5$ and $0 \le y \le 6$, and the global optimum is -2.345 at location $(x,y) = (4,\ 2)$. Like the case of Haupt function, two starting conditions are considered. In Fig. 8, the starting point is adopted as the center of entire design space and the radius of initial trust region is selected as 1/8 times the size of design space (=6/8). In Fig. 9, the starting point is adopted as the lower-left corner of design space and the radius of initial trust region is selected as the size of entire design space (=6). In case of the first starting condition adopted in Fig. 8, all of the procedures find the global optimum successfully. It seems that it is caused by the fact that Hosaki function is globally smoother than Haupt function.

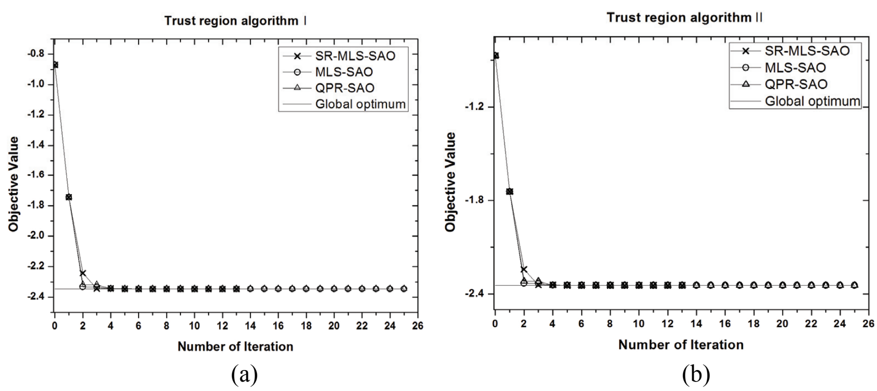


Figure 8: Iteration history for optimization of Hosaki function (1^{st} starting condition).

However, in case of the second starting condition, QPR-SAO (SAO with the quadratic polynomial regression meta-model) fails to find the global optimum as presented in Fig. 9b (Trust region algorithm II), whereas the global optimum is successfully found by SR-MLS-SAO and MLS-SAO procedures for both of trust region algorithms I and II.

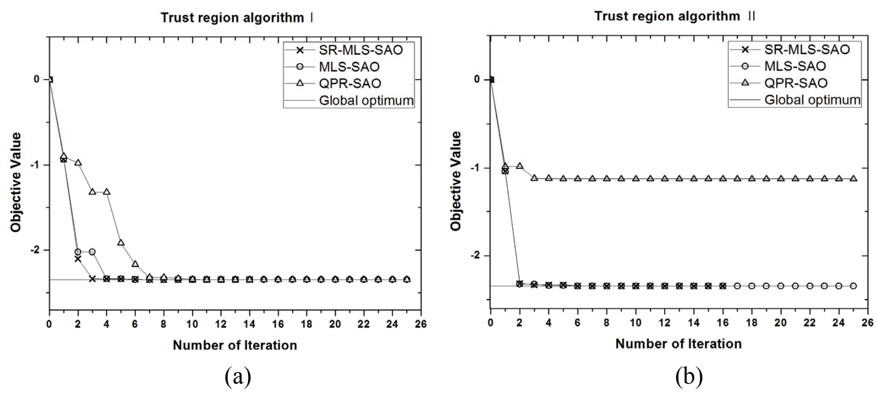


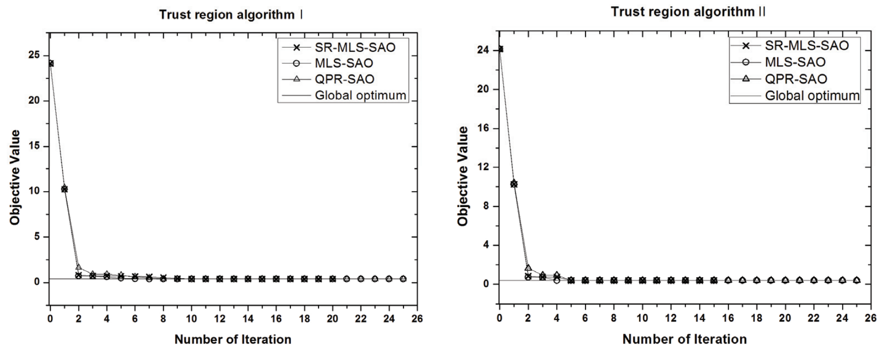
Figure 9: Iteration history for optimization of Hosaki function (2^{nd} starting condition).

3.3 *Optimization of Branin function*

At third, Branin function is optimized by the proposed sample-reusable SAO procedure. Branin function is written as follows [Branin (1972)].

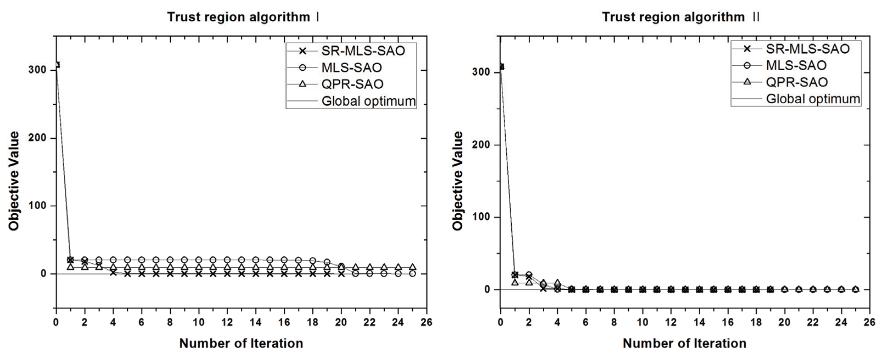
$$f^{(Branin)}(x,y) = \left(y - \frac{5x^2}{4\pi^2} + \frac{5x}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x) + 10 \tag{15}$$

Design space is chosen as $-5 \leq x \leq 10$ and $0 \leq y \leq 15$, and the global optimum is 0.3979 at three locations $(x,y) \in \{(-\pi,\ 12.25),\ (\pi,\ 2.25),\ (3\pi,\ 2.25)\}$. Like the former examples, two starting conditions are considered. In Fig. 10, the starting point is adopted as the center of entire design space and the radius of initial trust region is selected as 1/8 times the size of design space. In Fig. 11, the starting point is adopted as the lower-left corner of design space and the radius of initial trust region is the same as the size of entire design space. In Fig. 10, all of the solutions converge to the global optimum value, although the convergence rate of SR-MLS-SAO procedure is slightly slower than those of the others in Fig 10a. Under close investigation, it has been known that it is caused by inappropriate reduction of size of trust region in trust region algorithm. The effect of trust region algorithm will be investigated in section 4. In case of the second starting condition, QPR-SAO does not find the global optimum and MLS-SAO shows very slow convergence, whereas SR-MLS-SAO procedure gives the global optimum rapidly as shown in Fig. 11a. Based upon the results, it is identified that the proposed sample-reusable SAO procedure is more robust and reliable compared with the other SAO procedures.



(a) (b)

Figure 10: Iteration history for optimization of Branin function (1^{st} starting condition).



(a) (b)

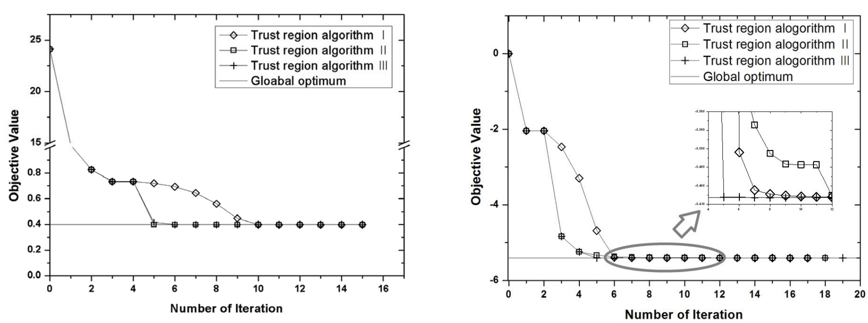
Figure 11: Iteration history for optimization of Branin function (2^{nd} starting condition).

4 Effect of Modification of Trust Region Algorithm

As observed in previous examples, the proposed sample-reusable SAO procedure is more robust and reliable compared with the other procedures. However, there is no guarantee that SR-MLS-SAO finds the global optimal point, since the SR-MLS-SAO inherently relies on the trust region method, which only guarantees the local minima convergence [Fletcher (1987)]. Further, the convergence rate may change according to the trust region algorithm. Therefore, the effect of change of trust region algorithm is investigated in this section. Trust region algorithm III described in section 2.1 is employed along with the proposed SR-MLS-SAO and the effect is investigated.

With the 1^{st} starting condition(same as Fig. 10), Branin function is optimized by SR-MLS-SAO with trust region algorithm III, and the result is compared with those obtained with trust region algorithms I and II in Fig. 12a. Comparison shows that the convergence rate could be accelerated according to the modification of trust region algorithm.

Under the 2^{nd} starting condition(same as Fig. 7), optimization of Haupt function is carried out by SR-MLS-SAO with trust region algorithm III, and the result is presented in Fig. 12b. It shows interesting result. The solution path of algorithm III is similar to that of algorithm II until 4 steps (see large picture), however after 4 steps, the path becomes similar to that of algorithm I which shows faster convergence than algorithm II after 5 iteration steps (see small picture). From the results, it can be confirmed that the algorithm III inherits the merits of both algorithms I and II. And it is identified that SR-MLS-SAO procedure with trust region algorithm III gives reliable solutions better than or similar to the trust region algorithms I and II. Based on the results, trust region algorithm III will be utilized along with the sample-reusable MLS meta-model in optimization of preliminary design of compound helicopter in the next section.



(a) Branin (1^{st} starting condition) (b) Haupt (2^{nd} starting condition)

Figure 12: Modification of trust region algorithm and its effect.

Figure 13: Push-type compound helicopter.

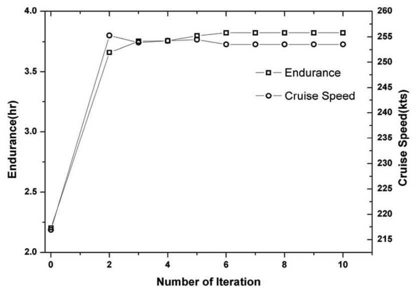
5 Optimization of Preliminary Design of Compound Helicopter

Compound helicopter is one of the promising VTOL(vertical take-off and landing) vehicle concepts, which may expand the mission area of conventional helicopter[Orchard, Newman (2003)]. In this work, push-type compound helicopter is considered. The push-type compound helicopter is equipped with additional wing and auxiliary propulsion unit as shown in Fig. 13.

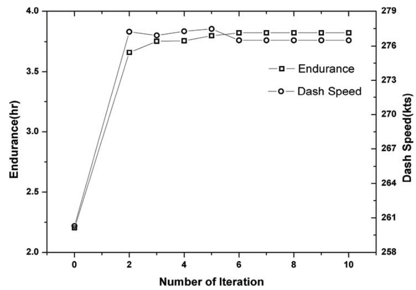
To investigate the expandability of possible mission area of compound helicopter, optimization of preliminary design of compound helicopter is carried out by the proposed sample-reusable SAO procedure. Additionally, the potential of the proposed sample-reusable SAO procedure is observed in practical point of view.

The baseline design parameters used for preliminary design are presented in Tab. 1. Design variables are presented in Tab. 2. Among the design variables, additional wing area and angle of attack of additional wing are distinct features of compound helicopter compared with the conventional helicopter.

In this work, to evaluate the performances of compound helicopter (such as endurance, cruise speed, dash speed, maximum range, and hovering capability) according to design variables, performance analysis program is utilized. In the program, required power for compound helicopter is calculated by the blade element theory [Prouty (1995)], and it is assumed that all thrust to the forward direction is provided by the auxiliary propulsion unit. Additionally, lift induced by fuselage is ignored.



(a) Endurance and cruise speed



(b) Endurance and dash speed

Figure 14: Optimization histories of preliminary design of compound helicopter.

Table 1: Baseline design parameters for preliminary design of compound helicopter.

Baseline design parameters	
Weight Baseline: the weight of rotorcraft except wings [lbs]	30000
Weight Fuel: maximum fuel weight [lbs]	6000
Flat Plate Drag area: frontal equivalent flat plate drag area [ft^2]	25
Horizontal Flat Plate Drag area: horizontal equivalent flat plate drag area [ft^2]	100
IRP: intermediate rated power [shp]	8000
MCP: maximum continuous power [shp]	6300
SFC: specific fuel consumption [lb/shp/Hr]	0.5

Table 2: Design variables selected for preliminary design of compound helicopter.

Design Variables	
R_R	Main Rotor Radius(ft)
R_C	Main Rotor Chord(ft)
R_{TV}	Main Rotor Tip Speed(ft/s)
W_A	Additional Wing Area(ft^2)
W_{AOA}	Angle of Attack for Additional Wing(deg)

At first, the endurance, cruise speed, and dash speed of compound helicopter are independently optimized, and those optimal values are utilized for multi-objective optimization. The obtained optimal values for endurance, cruise speed, and dash speed are denoted by f_E^*, f_C^*, and f_D^*, respectively. In multi-objective optimization, the following objective function $F^{(multi)}(\mathbf{x})$ and design region Ω are considered.

$$F^{(multi)}(\mathbf{x}) = \omega_E \left(\frac{f_E^* - f_E(\mathbf{x})}{f_E^*} \right)^2 + \omega_C \left(\frac{f_C^* - f_C(\mathbf{x})}{f_C^*} \right)^2 + \omega_D \left(\frac{f_D^* - f_D(\mathbf{x})}{f_D^*} \right)^2 \quad \text{for } \mathbf{x} \in \Omega \tag{16}$$

$$\Omega = \Big\{ \mathbf{x} : 20 \le R_R \le 40, 1.2 \le R_C \le 3,\ 450 \le R_{TV} \le 650, 0 \le W_A \le 400, 0 \le W_{AOA} \le 10 \Big\} \tag{17}$$

where $\mathbf{x} = [R_R,\ R_C,\ R_{TV},\ W_A,\ W_{AOA}]$ is the design variable. $f_E(\mathbf{x})$, $f_C(\mathbf{x})$, and $f_D(\mathbf{x})$ denote the endurance, cruise speed, and dash speed for the given design variable $\mathbf{x}$, respectively. Weighting factors for endurance, cruise speed, and dash speed are adopted as $\omega_E = 0.6$, $\omega_C = 0.2$, and $\omega_D = 0.2$, respectively.

Additionally, feasible design region is restricted by three conditions, considering the mission of compound helicopter. The conditions are as follows. (a) Maximum range should be longer than or equal to 400 NM. (b) Endurance should be longer than or equal to 2hr. (c) Hovering should be possible. Under these conditions, constraints are constructed as

$$g_i(\mathbf{x}) \le 0 \ (1 \le i \le 3) \tag{18}$$

In this work, the penalty method is utilized to handle the constraints. Through the penalty method, aforementioned multi-objective optimization problem could be written as shown below.

$$\mathbf{x}^{optimum} = ARG \left[\min_{\mathbf{x}\in\Omega} \left(F^{(multi)}(\mathbf{x}) + \alpha_p \sum_{i=1}^{3} [\max(g_i(\mathbf{x}),\ 0)]^2 \right) \right] \tag{19}$$

where the penalty parameter α_p is adopted as 10^4.

As described before, the endurance, cruise speed, and dash speed of compound helicopter are independently optimized to obtain f_E^*, f_C^*, and f_D^* at first. Trust region algorithm III is utilized along with 150 Latin hypercube sampling points. Iteration is terminated if the change of optimal value is less than 0.1% during consecutive 5 steps. For the three SAO procedures(SR-MLS-SAO, MLS-SAO, and QPR-SAO), the optimized results of f_E^*, f_C^*, and f_D^* are presented in Tab. 3. Once again, it is identified from the results that SR-MLS-SAO shows better performance than the other SAO procedures with no additional sampling cost.

With the optimal values f_E^*, f_C^*, and f_D^* obtained by SR-MLS-SAO procedure, multi-objective optimization is performed. In optimization, SR-MLS-SAO procedure is utilized along with trust region algorithm III, the same sampling points, and the same termination routine as before. The optimization results are presented in Tab. 3, and the optimization histories are presented in Fig. 14. From the results, it is identified that there are many possibilities to expand the mission area (such as rescue, reconnaissance, patrol or others) of conventional helicopter through increasing the endurance, cruise speed, and dash speed at the same time.

6 Conclusions

To fully utilize the information of previously sampled points in sequential approximate optimization (SAO) procedure, a sample-reusable SAO procedure is suggested. The SAO procedure utilizes a newly proposed sample-reusable MLS(moving least squares) meta-model along with the trust region algorithm. Under the nearness concept, the sample-reusable MLS meta-model extracts the valuable information about the system response not only from the pre-sampled points located in the current trust region, but also from the pre-sampled points near the current trust region. Consequently, it becomes possible to enhance the quality of meta-model for system response compared with conventional SAO procedures without additional expensive sampling cost.

Through the representative examples, the performance of the sample-reusable SAO procedure (SAO with sample-reusable MLS meta-model) is investigated, and the performance is compared with those of QPR-SAO (SAO with quadratic polynomial regression meta-model) and MLS-SAO (SAO with moving least squares meta-model). From the results, it is identified that the proposed sample-reusable SAO procedure gives more reliable solution than the others with no additional sampling cost, and it is less sensitive to starting conditions. Additionally, through the investigation of the effect of trust region algorithm, it is observed that modified trust region algorithm III could improve the convergence performance in the sample-reusable SAO procedure.

As a practical example, optimization of preliminary design of compound helicopter is carried out by using the proposed sample-reusable SAO procedure. From the optimization results, it is confirmed that it is possible to expand the mission area of conventional helicopter by using the concept of compound helicopter. Further,

Table 3: Optimization results for preliminary design of compound helicopter.

	Initial	Dash Speed Opt.			Cruise Speed Opt.			Endurance Opt.			Multi-obj Optimum
		SR-MLS	**MLS**	**QPR**	**SR-MLS**	**MLS**	**QPR**	**SR-MLS**	**MLS**	**QPR**	**SR-MLS**
Dash Speed(kts)	260.3	282.1	278.5	272	281.5	274.4	261.9	260.4	260.9	266.5	276.5
Cruise Speed(kts)	216.9	253.2	250.3	243.3	258.8	251.3	231.4	238.7	239.2	245.5	253.6
Endurance(hr)	2.202	2.964	2.6	2.95	3.442	3.365	3.046	5.201	5.175	3.933	3.823

it is identified that the proposed sample-reusable SAO procedure has a practical potential in engineering practice.

Acknowledgement: This work was supported by the Korea Research Foundation Grant (KRF-2008-013-D00019). Authors also appreciate the helpful comments from Dr. Ahn and Mr. Lee from the Georgia Institute of Technology.

References

Alexandrov, N. M.; Lewis, R. M.; Gumbert, C. R.; Green, L. L.; Newman, P. A. (2001): Approximation and model management in aerodynamic optimization with variable-fidelity models. *Journal of Aircraft*, vol. 38, no. 6, pp. 1093-1101.

Atluri, S. N.; Cho, J. Y.; Kim, H. S. (1999): Analysis of thin beams, using the meshless local Petrov-Galerkin method, with generalized moving least squares interpolations. *Computational Mechanics*, vol. 24, no. 5, pp. 334-347.

Atluri, S. N.; Han, Z.; Shen, S. (2003): Meshless Local Patrov-Galerkin (MLPG) approaches for weakly singular traction & displacement boundary integral equations. *CMES: Computer Modeling in Engineering & Sciences*, vol. 4, no. 5, pp. 507–517.

Atluri, S. N. (2004): *The Meshless Local Petrov-Galerkin (MLPG) Method.* Tech Science Press.

Atluri, S. N. (2005): *Methods of Computer Modeling in Engineering & the Sciences*, *Vol.* 1. Tech Science Press.

Bekey, G. A.; Ung, M. T. (1974): A comparative evaluation of two global search algorithms. *IEEE Trans. Systems, Man & Cybernetics*, vol. 4, no. 1, pp. 112-116.

Branin, F. H. (1972): Widely convergent methods for finding multiple solutions of simultaneous nonlinear equations. *IBM Journal of Research and Development*, vol. 16, no. 5, pp. 504-522.

Fang, K. T.; Wang, Y. (1994): *Number-theoretic Methods in Statistics*. Chapman & Hall.

Fang, K. T.; Wang, Y.; Bentler, P. M. (1994): Some applications of number-theoretic methods in statistics. *Statistical Science*, vol. 9, no. 3, pp. 416-428.

Fletcher, R. (1987): *Practical Methods of Optimization, 2nd Ed.* John Wiley & Sons.

Haupt, R. L.; Haupt, S. E. (1998): *Practical Genetic Algorithms.* John Wiley & Sons.

Ho, S. L.; Yang, S. Y.; Ni, P. H.; Wong, K. F. (2007): An adaptive interpolating MLS based response surface model applied to design optimizations of electromagnetic devices. *IEEE Transactions on Magnetics*, vol. 43, no. 4, pp. 1593-1596.

Hwang, H. S. (2000): *Model-based Cluster Analysis using Gaussian Processes and Lattice-based Latin Hypercube Designs*. Doctoral Thesis, Department of Computer Science and Statistics, Chonnam National University.

Johnson, M. E.; Moore, L. M.; Ylvisaker, D. (1990): Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, vol. 26, no. 2, pp. 131-148.

Jones, D. R. (2001): A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, vol. 21, no. 4, pp. 345-383.

Jones, D. R.; Schonlau, M.; Welch, W. J. (1998): Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, vol. 13, no. 4, pp. 455-492.

Lancaster, P.; Salkauskas, K. (1981): Surface generated by moving least squares methods. *Mathematics of Computations*, vol. 37, no. 155, pp. 141-158.

McDonald, D. B.; Grantham, W. J.; Tabor, W. L.; Murphy, M. J. (2007): Global and local optimization using radial basis function response surface models. *Applied Mathematical Modeling*, vol. 31, no. 10, pp. 2095-2110.

McKay, M. D.; Beckman, R. J.; Conover, W. J. (1979): A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics,* vol. 21, no. 2, pp. 239-245.

Myers, R. H.; Montgomery, D. C. (2002): *Response Surface Methodology, 2nd Ed*. John Wiley & Sons.

Orchard, M.; Newman, S. (2003): The fundamental configuration and design of compound helicopter. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 217, no. 6, pp.297-315.

Panda, S. S.; Manohar, C. S. (2008): Applications of meta-models in finite element based reliability analysis of engineering structures. *CMES: Computer Modeling in Engineering & Sciences*, vol. 28, no. 3, pp. 161-184.

Park, J. S. (1994): Optimal Latin-hypercube designs for computer experiments. *Journal of Statistical Planning and Inference*, vol. 39, no. 1, pp. 95-111.

Prouty, R. W. (1995): *Helicopter Performance Stability, and Control.* Krieger Pub. Co.

Rodriguez, J. F.; Renaud, J. E.; Watson, L. T. (1998): Convergence of trust region augmented Lagrangian methods using variable fidelity approximation data. *Structural Optimization*, vol. 15, nos. 3-4, pp. 141-156.

Sacks, J.; Welch, W. J.; Mitchell, T. J.; Wynn, H. P. (1989): Design and analysis of computer experiments. *Statistical Science*, vol. 4, no. 4, pp. 409-435.

Schmit, L. A., Jr.; Farshi, B. (1974): Some approximation concepts for structural synthesis. *AIAA Journal*, vol. 13, no. 5, pp. 692-699.

Sorensen, D. C. (1982): Newton's method with a model trust region modification. *SIAM Journal on Numerical Analysis*, vol. 19, no. 2, pp. 409-426.

Sovieski, I. P.; Manning, V. M.; Kroo, I. M. (1998): Response surface estimation and refinement in collaborative optimization. *AIAA Paper* 98-4753.

Tang, B. (1993): Orthogonal array-based Latin hypercubes. *Journal of the American Statistical Association*, vol. 88, no. 424, pp. 1392-1397.

Wang, G. G.; Shan, S. (2007): Review of metamodeling techniques in support of engineering design optimization. *Transactions of the ASME, Journal of Mechanical Design*, vol. 129, no. 4, pp. 370-380.

Ye, K. Q. (1998): Orthogonal column Latin hypercubes and their application in computer experiments. *Journal of the American Statistical Association*, vol. 93, no. 444, pp. 1430-1439.

Zadeh, P. M.; Toropov, V. V.; Wood, A. S. (2009): Metamodel-based collaborative optimization framework. *Structural Multidisciplinary Optimization*, vol. 38, no. 2, pp. 102-115.