IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 17, 2020, accepted October 2, 2020, date of publication October 19, 2020, date of current version October 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3032122

Modified Algorithms for Fast Construction of Optimal Latin-Hypercube Design

QINGYU WANG[1], TAKUJI NAKASHIMA[2], CHENGUANG LAI[3], HIDEMI MUTSUDA[2], TAIGA KANEHIRA[2], YASUFUMI KONISHI[4], AND HIROYUKI OKUIZUMI[4]

[1]Graduate School of Engineering, Hiroshima University, Higashihiroshima 7398527, Japan
[2]Graduate School of Advanced Science and Engineering, Hiroshima University, Higashihiroshima 7398527, Japan
[3]College of Vehicle Engineering, Chongqing University of Technology, Chongqing 400054, China
[4]Institute of Fluid Science, Tohoku University, Sendai 9808577, Japan

Corresponding author: Takuji Nakashima (nakashima@hiroshima-u.ac.jp)

This work was respectively supported in part by Japan Society for the Promotion of Science Grants-in-Aid for Scientific Research (JSPS KAKENHI) Grant Number 20K04286 and China Scholarship Council (CSC) Grant Number 201908500118.

ABSTRACT As accuracy of optimization can not be guaranteed without high-quality samples, the distribution of a finite number of evaluation points where experiments should be conducted in design space is an important issue, particularly when the experiment to obtain sample is expensive. To utilize limited number of evaluation points to represent the design space, optimal latin-hypercube design (OLHD), with considerable space-filling quality, is widely used as a methodology in design of experiments (DOE). However, OLHD generation requires significant time. This study focuses on further improvement of efficiency in generation of OLHD in terms of both time and latin-hypercube design (LHD) optimization. Two modified algorithms, namely the modified enhanced stochastic evolutionary (MESE) and translational propagation modified enhanced stochastic evolutionary (TPMESE) algorithms, based on existing algorithms, are proposed. The MESE algorithm is modified from the enhanced stochastic evolutionary (ESE) algorithm by using a new update method for "temperature," while the TPMESE algorithm optimizes the LHD via translational propagation (TPLHD) instead of optimizing a random LHD like the MESE algorithm does. Their performance is evaluated by comparison with several famous heuristic algorithms and each original algorithm using optimization tests of LHDs with various sizes. For all cases, proposed algorithms show better performance of convergence than other heuristic algorithms participated in our comparison. For large and medium LHDs, the MESE algorithm faster converges to a solution with the same level as original algorithm (ESE). For large LHDs, the TPMESE algorithm is the most time efficient algorithm in obtaining near-optimal or sufficient near-optimal designs.

INDEX TERMS Optimal latin-hypercube design, design of experiments, evolutionary algorithm, translational propagation.

I. INTRODUCTION

Optimization has become considerably popular in engineering design. In general, typical processes of optimization can be divided into three parts: sampling, construction of response surface, and search for optimal value [1]. Here, the response surface is an approximate model to establish the relationship between variables and the objective [2]. It can also be considered as a regression function to predict unknown space and filter numerical noise based on discrete samples. The selection of evaluation points in sampling is a critical step in obtaining a high-quality response surface in such a manner that the optimal value can be accurately searched in the next process. As most practical engineering problems are time-consuming, it is unfeasible to conduct a lot of experiments. Therefore, effectively utilizing finite evaluation points to exhibit the properties of design space is a good choice to reduce experimental costs. The methodologies known as design of experiments (DOE) are widely used to assign locations for evaluation points. In these methodologies, the latin-hypercube design (LHD), proposed by Michael *et al.* [3] and Iman and Conover [4], has interesting characteristics, such as non-collapsing features and orthogonality [5]. If there is a design with n_p points and n_v dimensions, LHD cuts every dimension into n_p equal levels to satisfy the main property of LHD, one

The associate editor coordinating the review of this manuscript and approving it for publication was Ran Cheng.

point per level. However, because of the random permutation of evaluation points in LHD, their uniform distribution over the entire region can not be guaranteed. In literatures, the space-filling quality of the design is considered importantly to describe the entire design space. Therefore, many studies used different methods to optimize the space-filling quality of LHD. For example, Kenny *et al.* [6] utilized the columnwise–pairwise (CP) algorithm proposed in literature [7] to generate optimal symmetrical LHD. Fang *et al.* [8] adopted the threshold accepting (TA) algorithm in constructing optimal LHD (OLHD). Bates *et al.* [9] presented a permutation genetic algorithm (PermGA) to search an optimal LHD. Grosso *et al.* [5] developed the iterated local search (ILS) algorithm for LHD optimization. In particular, Jin *et al.* [10] proposed the enhanced stochastic evolutionary (ESE) algorithm, which has significant capability for LHD optimization through balancing global exploration and local exploitation. It should be noted that there are numerous possible solutions for a design with high dimensions and many evaluation points. Searching for an optimal solution for such design consumes considerable time. Hence, the balance between time costs and optimality of solution in the global optimization of LHD interests and challenges researchers. Thus, many outstanding efforts for improving efficiency in construction of LHDs with high space-filling quality were made, which include enhancement of enhanced stochastic evolutionary (EESE) algorithm (Chantarawong *et al.* [11]), successive local enumeration (SLE) algorithm (Zhu *et al.* [12]), particle swarm optimization (PSO) algorithm (Chen *et al.* [13]), sequencing optimization based on simulated annealing (SOBSA) algorithm (Pholdee and Bureera [14]), a new DOE framework based on PermGA (Kianifar *et al.* [15]), PermGA based on chromosome-length-expansion (CLE) scheme (Mahmoudi and Zimmermann [16]), slice latin-hypercube design (SLHD) (Ba *et al.* [17]), maximum projection design (Joseph *et al.* [18] and Joseph [19]), sequential-successive local enumeration (S-SLE) algorithm (Long *et al.* [20]), inflate, expand and stack (IES) algorithm (GuiBan *et al.* [21]), an efficient method for constructing space-filling and near-orthogonality Sequential LHD (Wu *et al.* [22]), a novel extension algorithm (Li *et al.* [23]), maximin distance latin squares and related latin-hypercube design based on Costas arrays and the Welch, Gilbert and Golomb methods (Xiao and Xu [24]) and local search-based genetic algorithm (LSGA) (Shang *et al.* [25]). Additionally, in publications, we noticed a quite efficient algorithm, the latin-hypercube via translational propagation (TPLHD), was developed by Viana *et al.* [26] to faster construct a near high-quality design. According to the paper, this is suitable as an initial design for typical optimization using optimization algorithms.

However, there are some important conclusions in relative literatures which deserve reviewing. Damblin *et al.* [27] compared the performance of simulated annealing (SA) and ESE algorithms in the construction of LHD with good space-filling quality. Husslage *et al.* [28] made a comparison of simulated annealing (SA), ESE and PermGA algorithms and the results showed that the ESE algorithm found better results than SA and PermGA algorithms for almost all of the cases. Moreover, the performance of ESE algorithm for establishment of OLHD with high space-filling quality was further validated in literatures [14], [22], [23] and [25] through comparing with SOBSA, GA, SLE, SLHD, LSGA and a novel extension algorithms. The results revealed that the ESE algorithm is a significantly efficient and robust algorithm for optimizations of LHDs within 10 dimensions.

This study focues on the development of an efficient technology to generate an OLHD with high space-filling quality, which matches the expectation of an issue in engineering optimization with considerably experimental costs. Considering the significant capability of ESE algorithm, an optimization algorithm for fast construction of OLHD is proposed, which is modified from ESE by introducing a new update method of the threshold ''temperature'' to enhance efficiency and maintain sufficient global searching capability. Furthermore, our proposed algorithm is combined with the TPLHD to directly optimize a near high-quality design instead of a random LHD for further acceleration of search for a sufficient near-optimal LHD in the beginning of the optimization process, particularly in the case of large size LHDs. It can be predicted that these efforts are significantly beneficial to efficiently obtain high-quality samples in practical optimization problems. The proposed algorithms are performed in optimization tests with various LHD sizes. The test results are compared with other well-known algorithms, including PermGA, ILS and LSGA, and the original algorithms. Then, their performance is demonstrated.

II. FUNDAMENTAL TECHNOLOGY

In this Section, some important fundamental techniques, including optimality criterion used in optimization (Subset A), TPLHD (Subset B) and ESE algorithms (Subset C), are reviewed, so that the study can be more clearly understood.

A. CRITERION OF JUDGEMENT FOR OPTIMIZATION

As in other optimization problems, some criteria were proposed by researchers to evaluate the performance of space-filling for LHD optimization. ϕ_p is a popular criterion improved by Morris and Mitchell [29] to be used in estimating uniformity of LHD space. It is calculated as follows:

$$\phi_p = [\sum_{i=1}^{n_p-1} \sum_{j=i+1}^{n_p} d_{ij}^{-p}]^{1/p} \tag{1}$$

where d_{ij} is the distance between two random points and n_p is the number of points. In the LHD optimization, maximizing the distance between two random points satisfies the requirement of the spacing-filling property. From the expression, maximizing the distance between points is equivalent to minimizing the ϕ_p value. d_{ij} can be expressed as follows:

$$d_{ij} = [\sum_{k=1}^{n_v} |x_{ik} - x_{jk}|^t]^{1/t} \quad (2)$$

where x_{ik} and x_{jk} indicate the kth component of i and j points, respectively. Normally, $p = 50$ and $t = 1$ are suitable settings for most situations in LHD optimization [10]. In this study, the performance of LHD optimization is evaluated using ϕ_p.

B. TRANSLATIONAL PROPAGATION ALGORITHM

TPLHD is an algorithm that aims to fill up the space of LHD by a translational process from small seed design.

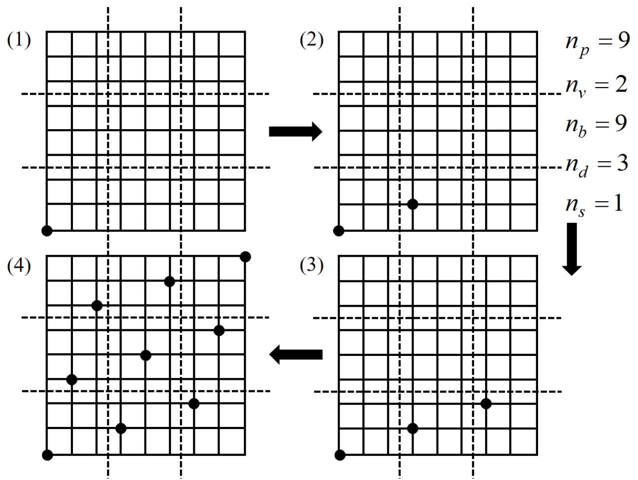

FIGURE 1. Process of translational operation in TPLHD. After the stratification of 9 blocks for LHD space, the point placed in the original coordinates translates in the horizontal direction until the situation seen in step 3 occurs. Then, the three points, as the new seed, repeat the translational process in the vertical direction.

Figure 1 shows the details of this process using the construction of a 9 × 2 TPLHD ($n_p = 9$ points, $n_v = 2$ dimensions) that is started from a 1 × 2 ($n_s = 1$ point of seed) seed as an example. Initially, the entire design space is distributed into n_b blocks. The number n_b can be calculated by the following expression:

$$n_b = n_p / n_s \quad (3)$$

In sequence, the seed point located at the original coordinates translates in the first dimension (horizontal direction) per n_d levels until all the divisions in the first dimension are filled by seeds, where n_d divisions are expressed as:

$$n_d = n_b^{1/n_v} \quad (4)$$

In addition, to ensure that there is a single point per level, one level should be simultaneously translated through another dimension (vertical direction in this example) in the aforementioned seed translation process. When the divisions of the first dimension are filled by seeds, this process is repeated in the second dimension. To construct a TPLHD of any size, particularly when the number of points in the TPLHD is not the required number, the resizing process is employed to

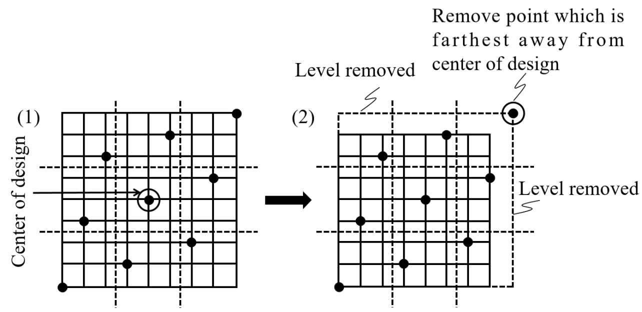

FIGURE 2. Resizing process in TPLHD. To construct an 8 × 2 LHD using the Translational Propagation algorithm, a larger sized TPLHD (9 × 2) is initially constructed. Then, distance between the center of design and every point is evaluated. Finally, the point that is farthest away from the center of design and its corresponding levels are removed.

remove points and their corresponding levels from a TPLHD of larger size according to descending permutation of distances between the center of the TPLHD and every point in it. Figure 2 exhibits the resizing process.

It is notable that TPLHD has high efficiency as only the translational process is performed without any mathematic calculation. Thus, comparing it with other optimization algorithms, the TPLHD focuses on the improvement of efficiency for the construction of a near high-quality LHD, not the best one. In contrast, different TPLHD spacing-filling properties can be obtained using different seed points for the construction of TPLHD [26]. Considering its low time cost, it is possible to select the best TPLHD from several TPLHDs constructed using different seeds. To build an initial near high-quality LHD in the global optimization based on the new algorithm we modified, the seed points from 1 to 5 are used to construct different TPLHDs of the required points.

TABLE 1. Pseudo code for TPLHD procedure.

Line	Construction procedure for TPLHD
1	Set n_p, n_v.
2	For n_s = 1 to 5
3	If n_d is an integer
4	$n_b = n_p / n_s$
5	Else
6	$n_b = \text{ceil}(n_d)^{n_v}$
7	Recalculate points number n_p^* based on the new n_b.
8	End If
9	Generate initial TPLHD.
10	If $n_p^* > n_p$
11	Resizing process to remove points from initial TPLHD.
12	End If
13	End For
14	Pick the best TPLHD based on min value of ϕ_p.

Then, the best one is selected as the initial LHD for global optimization on the basis of comparison via ϕ_p criterion. The procedure is listed in Table 1.

C. ENHANCED STOCHASTIC EVOLUTIONARY ALGORITHM

The ESE algorithm is derived from a normal stochastic evolutionary algorithm [30]. It involves two loops, the inner and outer loops, to automatically update "temperature" in accordance with the degree of improvement and perform the classical element-exchange operation within a column for LHD. In the inner loop, J distinct element-exchanges within one column are implemented based on current design X in every iteration. In the next step, the best design X_{try} is selected from the J randomly distinct element-exchanges. If the X_{try} meets the requirement of the acceptance criterion, the current design X is replaced by X_{try}. Then, the design X and the current best design X_{best} are compared based on the ϕ_p criterion. Finally, the best design between X and X_{best} is chosen. This process is performed until M iterations are completed. The acceptance criterion mentioned is as follows:

$$f(X_{try}) - f(X) \leq T_h \cdot \mathrm{rand}(0, 1) \tag{5}$$

where $f(X_{try})$ and $f(X)$ are the values of ϕ_p for X and X_{try} design, respectively. T_h is the value of "temperature" as the threshold. The function rand(0,1) selects a random number from 0 to 1.

The main function of the outer loop is to control the change of "temperature" T_h. According to judgement of tolerance, the outer loop can be divided into improvement and exploration processes. Then, T_h is adjusted using different equations based on the acceptance n_{acpt}/M and improvement n_{imp}/M ratios updated in the inner loop.

The procedure for ESE algorithm is shown in Table 2 and Figure 3 (the figure was drawn based on literature [10]). For the settings of some basic parameters, the author of the original paper suggests that M and J are set as $M = 2n_e m / J$ and $J = n_e / 5$, respectively. In addition, it is reasonable that J and M should not be larger than 50 and 100, respectively. Here, n_e is the number of all possibilities for element-exchanges in one column.

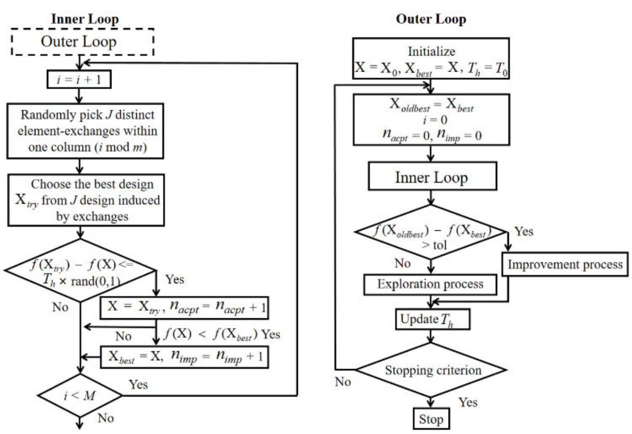

FIGURE 3. Flowchart of the ESE algorithm. The figure was drawn based on literature [10].

In the ESE algorithm, the main property is the update of "temperature" in the outer loop. As mentioned, the update process can be divided into improvement and exploration processes. For the improvement process, if the acceptance ratio n_{acpt}/M is larger than a threshold such as 0.1 and the improvement ratio n_{imp}/M, the "temperature" will be reduced through the equation $T_h = \alpha_1 T_{h\_old}$. When the acceptance ratio n_{acpt}/M is larger than a small value (e.g., 0.1) but equal to the improvement ratio n_{imp}/M, T_h is maintained. Otherwise, T_h is increased by equation $T_h = T_{h\_old}/\alpha_1$. The improvement process aims to promote the fast convergence of the algorithm close to a local solution. In contrast, T_h is decreased by $T_h = \alpha_2 T_{h\_old}$ when n_{acpt}/M is larger than a large percentage like 0.8 in the exploration process. While the n_{imp}/M is smaller than a small threshold (e.g., 0.1), T_h is increased by $T_h = T_{h\_old}/\alpha_3$. Unlike the improvement process, this process focuses on escaping from the local solution to find an optimal design. According to the original literature, α_1, α_2 and α_3 are set to 0.8, 0.9, and 0.7, which can adequately work for all tests.

TABLE 2. Pseudo code for ESE procedure.

Line	ESE procedure for inner loop
1	For $i = 1$ to M
2	Pick the best X_{try} from the process of J distinct element-exchanges within a column (i mod m).
3	If $f(X_{try}) - f(X) \leq T_h \cdot \mathrm{rand}(0,1)$
4	$X = X_{try}$
5	$n_{acpt} = n_{acpt} + 1$
6	If $f(X) < f(X_{best})$
7	$X_{best} = X$
8	$n_{imp} = n_{imp} + 1$
9	End If
10	End If
11	End For
Line	ESE procedure for outer loop
1	Initialize $X = X_0$, $X_{best} = X$, $X_{oldbest} = X_{best}$, $T_h = T_0$ based on initial X_0 design and "Temperature" T_0.
2	While stopping criterion is not satisfied
3	$X_{oldbest} = X_{best}$, $n_{acpt} = 0$, $n_{imp} = 0$
4	Perform inner loop.
5	If $f(X_{oldbest}) - f(X_{best}) > tol$
6	Perform improvement process to update T_h.
7	Else
8	Perform exploration process to update T_h.
9	End If
10	End While

III. MODIFIED ALGORITHMS FOR OPTIMIZATION OF LHD BASED ON TPLHD AND ESE ALGORITHMS

To overcome certain shortcomings in the ESE algorithm and further accelerate its convergence, the modified enhanced stochastic evolutionary (MESE) algorithm is proposed. In this section, we firstly discuss the details of MESE algorithm (Subset A). In the following, another algorithm combining MESE and TPLHD algorithms to further shorten the computational time from a poor starting design to a near high-quality design is elaborated (Subset B).

A. MODIFIED ENHANCED STOCHASTIC EVOLUTIONARY ALGORITHM

Similar to the ESE algorithm, the MESE algorithm also contains two loops (inner and outer loops). The process in the inner loop is the same; the main modification occurs in the outer loop. Two updating processes involving "temperature" T_h, the improvement and exploration processes, in the ESE algorithm, are combined in the MESE algorithm. In addition, to meet the complicated requirements of T_h updating at different stages in the optimization process, some scale factors of variable step length are used to replace constant factors such as α_1, α_2 and α_3 in the ESE algorithm. As the performance shows in ESE tests, "temperature" can be adaptively adjusted with the implementation of algorithm. Constant scale factors easily lead to oversized or undersized skip distance; this causes the loss of possible intermediate solutions that can generate a better result after several evolutions. The loss of intermediate solutions decelerates the convergence and increases the processing time as the algorithm needs to search again. In contrast, the improvement process is usually performed during the initial stage of optimization because of adequate diversity of individuals based on monitored values. In other words, T_h quickly decreases to find a local solution by running the improvement process at the beginning and vibrating during the rest of the period based on the exploration process.

Considering the disadvantages and properties of the update method for T_h, the new update method of T_h in MESE aims to inherit the basic properties of T_h shifted and employ scale factors of variable step length to increase the searching for intermediate solutions. For the new update method, the T_h is reduced if the acceptance ratio n_{acpt}/M exceeds a large percentage C_1. In this case, unlike in the ESE case, the scale factor controlling the reduction of T_h is not a constant but an adaptive value that will be updated with the acceptance ratio n_{acpt}/M as in the following expression:

$$T_h = T_{h\_old} \times (0.9 - \beta_1^{((1-C_1)/(n_{acpt}/M-C_1))^{n_1}}) \tag{6}$$

In Eq. (6), if the acceptance ratio n_{acpt}/M tends to 1, the equation changes to $T_h = (0.9 - \beta_1) \times T_{h\_old}$. The reduction of T_h can be approximately treated as the reduction of T_h in the improvement process of the ESE algorithm at the beginning of optimization through a small factor. When the acceptance ratio n_{acpt}/M is close to a threshold C_1, the scale factor is increased to 0.9 to provide a slight reduction of T_h. In particular, an exponent n_1 is applied to control the speed of shift for the scale factor. β_1, C_1 and n_1 are set to 0.1, 0.8 and 4, respectively, which can obtain good results for most tested problems. T_h is increased with different speeds if the acceptance ratio n_{acpt}/M is less than a value C_2, say, 0.2 and no better design is available, which means $n_{imp} = 0$. The update method has the same format as Eq. (6). Therefore, if the acceptance ratio n_{acpt}/M approaches C_2, the scale factor is adjusted to $0.7+\beta_2$ to provide a slow increase of T_h. The scale factor can also be changed to 0.7 to accelerate the increase of T_h and rapidly improve the diversity of the individual. The update method described is expressed as follows:

$$T_h = T_{h\_old}/(0.7 + \beta_2^{(1+(M/n_{acpt}-1)\times(1-n_{acpt}/M/C_2))^{n_2}}) \tag{7}$$

where β_2 and n_2 are 0.2 and 0.125, respectively, in accordance with test results.

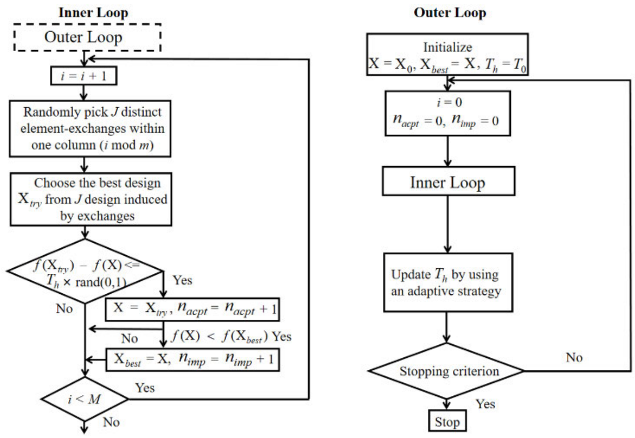

FIGURE 4. Flowchart of MESE algorithm. In the MESE algorithm, the outer loop is simplified in one integrative process to update "temperature".

TABLE 3. Pseudo code for outer loop of MESE algorithm.

Line	MESE procedure for outer loop
1	Initialize $X = X_0$, $X_{best} = X$, $T_h = T_0$ based on initial X_0 design and "Temperature" T_0.
2	While stopping criterion is not satisfied
3	$n_{acpt} = 0$, $n_{imp} = 0$
4	Perform inner loop.
5	If $(n_{acpt}/M) \geq C_1$
6	Reduce T_h.
7	Else If $(n_{acpt}/M) \leq C_2$ and $n_{imp} = 0$
8	Increase T_h.
9	Else If $C_2 < (n_{acpt}/M) < C_1$ and ($f(X) > S \times f(X_{best})$ or $n_{imp} = 0$)
10	Continue to reduce T_h.
11	End If
12	End While

When there are still many replacements to be conducted between design X_{try} and current design X to consume considerable computational time, the T_h continues to slowly decrease by a simple expression $T_h = \alpha T_{h\_old}$ to further promote convergence. Specifically, this operation is performed if the acceptance ratio n_{acpt}/M is within the range from C_2 to C_1 and satisfies one of the conditions: at least one improvement design is produced ($n_{imp} \neq 0$) or the current design X is worse than the current best design X_{best} over restriction, such as $f(X) > S \times f(X_{best})$. The reasonable scale factor α and restriction S could be 0.9 and a value which is no larger than 1.015, respectively. For other situations, T_h is maintained. Figure 4 shows the procedure of MESE, while Table 3 lists the procedure for the new outer loop.

B. SERIES BETWEEN TPLHD AND GLOBAL OPTIMIZATION ALGORITHM

In the past, LHD optimization usually started from a random LHD. Therefore, the convergence depended on the quality of the initial LHD. Thus, if the space-filling quality of the initial LHD is poor, the algorithm spends more time converging to a local solution in the optimization until the optimal solution is found. As there is only one final optimal result for the same size and range of every dimension of LHD, it is possible to accelerate convergence by shortening the time from a poor starting design to a near high-quality design. However, two preconditions should be satisfied before achieving this acceleration. The first one is that a near high-quality design can be generated fast, which implies its time cost should be shorter than the time from a random LHD to the equal-quality solution optimized by a global optimization algorithm. Second, the algorithm needs to have sufficient capability to rapidly escape from the local solution.

In Section II, an efficient algorithm, TPLHD, which can rapidly generate a near high-quality solution via translational operation, was illustrated. According to practical tests, generating a TPLHD with large size, such as 100 × 10, only requires several seconds, while it has almost no time costs for constructions of smaller size TPLHDs. Moreover, its time cost is shorter than the optimization process started from a random LHD to equal-quality LHD, regardless of the MESE or ESE used in the optimization. Thus, the combination of TPLHD and MESE or ESE seems a good choice to improve efficiency of optimization in initial stage. Therefore, in this paper, a new algorithm, namely translational propagation modified enhanced stochastic evolutionary (TPMESE) algorithm, is proposed to optimize LHD. As indicated in the name, this algorithm uses MESE to directly optimize a near high-quality design generated by TPLHD. Correspondingly, a TPESE algorithm, starting from TPLHD to use ESE performing optimization, is also applied to compare with TPMESE. Our motivation for developing this algorithm is to emphasize high efficiency at the beginning stage for a near-optimal or sufficient near-optimal designs, particularly when the time cost is the first concern.

IV. RESULTS AND DISCUSSION

In this section, the performance of proposed algorithms is evaluated through the experiments. Meanwhile, the analyses with respect to computational complexities and mechanisms of proposed algorithms acted in the tests are offered.

A. EXPERIMENTAL SETTING

To confirm the performance of the TPMESE and MESE algorithms, the proposed algorithms and each original ones were compared with several famous algorithms, including PermGA, ILS and LSGA. Such comparison aimed to confirm the convergence performance of the proposed algorithms and each original ones with respect to other heuristic algorithms for tested cases. Later, the performance of TPMESE, MESE, TPESE and ESE algorithms in the whole period of global convergence would be compared. Different from the comparison with other heuristic algorithms, this comparison was devoted to show the specific improvement of TPMESE and MESE with respect to original algorithms. All of comparisons were performed on three tested classifications based on LHDs sizes in optimization. Here, ESE and MESE algorithms start the optimization process from random LHD, while the remaining algorithms start from TPLHD. Regarding the tested classifications, two small (30 × 3 and 40 × 4), one medium (50 × 5), and two large size LHDs (60 × 6 and 100 × 10) were used. For the range of each variable, we kept it the same as that in literature [10].

During the experiment, all of algorithms would repeat the LHD optimization with different sizes for 100 times. For all tests, the ϕ_p criterion was used to evaluate the property of space-filling for LHD. Additionally, in comparing ESE and its variants, a global stopping criterion was used to indicate the time in which optimization can be treated as global convergence. If no better solution is generated within 1000 generations, global convergence can be regarded as reached [28]. Here, generation means the number of both outer and inner loops finished. To set parameters for TPMESE and MESE algorithms, the proposed algorithms were used to generate OLHD using the tests of 30 × 3, 50 × 5 and 100 × 10 LHDs for different values of parameters. For each combination of values of parameters, each simulation was repeated 10 times. We have to commit that in this stage the focus was not on the exploration of the optimal values of parameters for our algorithms. Therefore, we simply set β_1, β_2, n_1, n_2 and α to [0.1, 0.2], [0.1, 0.2], [2, 2.5, 3, 3.5, 4], [0.125, 0.25, 0.5] and [0.8, 0.85, 0.9, 0.95], respectively. In tests, each simulation was evaluated 0.5 million for ϕ_p criterion. The suitable values of parameters for most of tested cases were already described in Section III, except the 100 × 10 LHD. Here, β_1 was 0.2, while n_1, n_2 and α were 2.5, 0.5, 0.95, respectively. Regarding the parameter settings of other algorithms, we kept the same values suggested in Section II for ESE and TPESE algorithms. In addition, the initial value of "temperature" T_0 for both ESE and the three variants was calculated as $0.005 \times \phi_p$, based on the initial design. The tolerance in ESE was set as 0.0001 [11].

According to publication [31], population sizes were 20 × dimensions for small size LHDs, 10 × dimensions for medium and large size LHDs, respectively, in PermGA, while elite size, crossover and mutation rates were 5, 0.8, 0.05 [15], respectively. For LSGA, the author of literature [25] suggested that population number P, mutation probability p_m, parameters $p_{\max}$, $p_{\min}$ and distance ratio c were 10, 0.2, 0.3, 0.01 and 0.5, respectively, while there was no parameter settings for ILS.

To ensure the results of comparative investigations are statistically significant, t-test was used to compare mean values of ϕ_p criterion obtained in repeated simulations for all of tests according to literature [10]. In hypothesis testing, p-value was applied to evaluate the level of statistically significant difference. Here, the standard of p-value was set to 0.025%. In other words, if the p-value is smaller than the standard set between two groups of comparative samples, the null hypothesis (meaning values are equal) can be rejected. To help with clearer illustration in the following, symbols "algorithm-algorithm" or "algorithm-algorithm*" (e.g. "TPMESE-MESE*") were used to indicate the p-value was computed based on two groups of independent results from corresponding algorithms, where marker "$*$" denotes p-value is smaller than our standard. In contrast, no maker indicates p-value is larger than our standard. Particularly, symbol "algorithm-" indicates all p-values among ones and the remaining algorithms. To keep fair in comparisons, all of tests were conducted on a computer with 32GB RAM, eight-core CPU with clock speed of 3.60 GHz (Core i7-9700K) using MATLAB R2019b.

B. ANALYSIS OF COMPUTATIONAL COMPLEXITY

In the optimization of LHD, objective function will be repeatedly calculated whenever a new design is generated. Taking the ϕ_p criterion such Eq. (1) as an example, re-evaluation of ϕ_p criterion has to repeatedly evaluate the distances of all evaluation points and p-powers. Thus, re-evaluation of optimality criterion in optimization is significantly time-consuming [10], especially for the distance matrix of evaluation points in large size.

Considering the main computational costs came from re-evaluation of ϕ_p criterion, Here, we evaluated the computational complexity for different algorithms in the process of one evaluation of ϕ_p criterion. For our proposed algorithms, TPMESE and MESE algorithms, whose process is the same as ESE in inner loop, an evaluation of ϕ_p criterion for both ESE and its three variants will take $O(n_p^2 n_v)$ for the calculations of inter-site distances and $O(n_p^2 \log_2 p)$ for the computation of p-powers, respectively. In general, the overall computational complexities of TPMESE, MESE, TPESE and ESE algorithms are all $O(n_p^2 n_v) + O(n_p^2 \log_2 p)$. Similar with ESE algorithm, the ILS will also take $O(n_p^2 n_v) + O(n_p^2 \log_2 p)$. Regarding the LSGA and PermGA algorithms, the LSGA algorithm will additionally take $O(n_p n_v)$ and $O(n_v)$ for Modified Order Crossover (MOX) and Probabilistic Mutation operators, respectively, to generate a new design before each evaluation of ϕ_p criterion, while the PermGA algorithm requires extra $O(n_p)$ for Cycle Crossover operator in an evaluation of ϕ_p criterion. Thus, there are totally $O(n_p n_v + n_v + n_p^2 n_v + n_p^2 \log_2 p)$ and $O(n_p + n_p^2 n_v + n_p^2 \log_2 p)$ for LSGA and PermGA algorithms, respectively, in each evaluation of ϕ_p criterion. When we pick the term which grows the fastest, their computational complexities are both bounded by $O(n_p^2 n_v) + O(n_p^2 \log_2 p)$. Therefore, these algorithms have the same overall computational complexity as each other in the process of an evaluation of ϕ_p criterion.

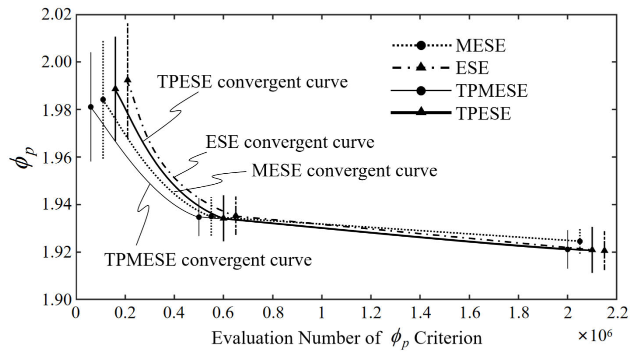

FIGURE 5. Comparison of optimization performance of different algorithms with 30 × 3 LHD. In this figure, every group of lines includes 4 lines gathered together. Every line has the same evaluation number of optimality criterion within one group. The mean values of the optimality criterion and the convergent curve are lower and the optimization converges faster. The gradient of the convergent curve indicates convergent speed; a steeper curve implies higher convergent speed.

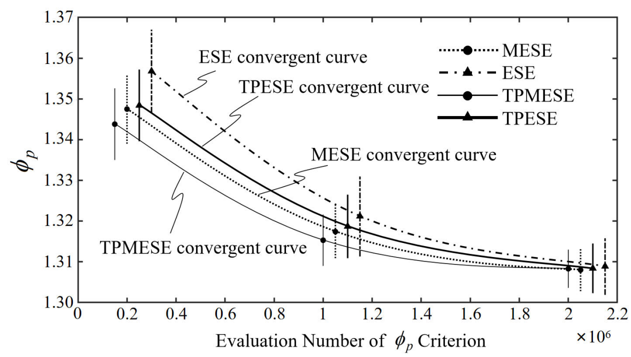

FIGURE 6. Comparison of optimization performance of different algorithms with 40 × 4 LHD.

C. RESULTS AND DISCUSSION OF COMPARATIVE INVESTIGATIONS

To illustrate the performance of TPMESE and MESE algorithms, we compared the mean values of the ϕ_p criterion and the corresponding standard deviations (std) for LHDs with different sizes and evaluation numbers of ϕ_p criterion in optimization, as shown in Tables 4 and 6, while Table 5 shows the mean values of computational time of different heuristic algorithms to reach the same near-optimal value of ϕ_p criterion. Particularly, the global optimal values of the ϕ_p criterion obtained after reaching global convergence in the 100 time tests are also listed in table 6, when our proposed algorithms were compared with ESE and TPESE algorithms. Figures 5 to 9 provide a distinct visualization of the optimization performance of the proposed algorithms with different LHD sizes. In these figures, the markers indicate the mean values at evaluation numbers of ϕ_p criterion, while the distances of vertical lines up and below the points are the standard deviations (std). It should be noted that equal intervals between any two adjacent vertical lines are maintained at every evaluation number of ϕ_p criterion we focused on, enabling a clear visualization of data. In other words, a group of different vertical lines gathered together

TABLE 4. Numerical results for optimization of LHDs with different sizes using different heuristic algorithms.

Size of LHD	Algorithms	Set1		Set2	
		#Evaluations	Mean (Std)	#Evaluations	Mean (Std)
30×3	MESE	50000	1.9915 (0.0265)	500000	**1.9350 (0.0081)**
	ESE	50000	1.9994 (0.0252)	500000	**1.9353 (0.0081)**
	TPMESE	50000	**1.9834 (0.0211)**	500000	**1.9347 (0.0080)**
	TPESE	50000	1.9912 (0.0239)	500000	**1.9342 (0.0097)**
	ILS	50000	2.0067 (0.0175)	500000	1.9675 (0.0163)
	LSGA	50000	2.0488 (0.0294)	500000	2.0216 (0.0250)
	PermGA	50000	2.1298 (0.0316)	500000	2.0349 (0.0313)
40×4	MESE	120000	1.3519 (0.0095)	1000000	1.3174 (0.0065)
	ESE	120000	1.3585 (0.0110)	1000000	1.3212 (0.0099)
	TPMESE	120000	**1.3474 (0.0076)**	1000000	**1.3153 (0.0063)**
	TPESE	120000	1.3516 (0.0080)	1000000	1.3187 (0.0078)
	ILS	120000	1.3708 (0.0089)	1000000	1.3522 (0.0079)
	LSGA	120000	1.3934 (0.0150)	1000000	1.3810 (0.0147)
	PermGA	120000	1.4496 (0.0135)	1000000	1.3868 (0.0148)
50×5	MESE	120000	1.0244 (0.0066)	2000000	**0.9871 (0.0036)**
	ESE	120000	1.0311(0.0067)	2000000	0.9912 (0.0049)
	TPMESE	120000	**1.0209 (0.0059)**	2000000	**0.9881 (0.0043)**
	TPESE	120000	1.0276 (0.0061)	2000000	0.9923 (0.0055)
	ILS	120000	1.0378 (0.0083)	2000000	1.0176 (0.0056)
	LSGA	120000	1.0538 (0.0079)	2000000	1.0387 (0.0081)
	PermGA	120000	1.1061 (0.0090)	2000000	1.0423 (0.0091)
60×6	MESE	120000	0.8265 (0.0047)	2000000	**0.7936 (0.0029)**
	ESE	120000	0.8284 (0.0049)	2000000	0.7976 (0.0034)
	TPMESE	120000	**0.8240 (0.0044)**	2000000	**0.7931 (0.0031)**
	TPESE	120000	0.8267 (0.0049)	2000000	0.7964 (0.0033)
	ILS	120000	0.8326 (0.0057)	2000000	0.8118 (0.0039)
	LSGA	120000	0.8465 (0.0050)	2000000	0.8222 (0.0054)
	PermGA	120000	0.9088 (0.0056)	2000000	0.8336 (0.0050)
100×10	MESE	1000000	0.4466 (0.0011)	2000000	0.4439 (0.0010)
	ESE	1000000	0.4490 (0.0013)	2000000	0.4450 (0.0009)
	TPMESE	1000000	**0.4459 (0.0008)**	2000000	**0.4435 (0.0008)**
	TPESE	1000000	0.4481 (0.0012)	2000000	0.4446 (0.0010)
	ILS	1000000	0.4484 (0.0010)	2000000	0.4457 (0.0009)
	LSGA	1000000	0.4531 (0.0014)	2000000	0.4494 (0.0012)
	PermGA	1000000	0.5053 (0.0027)	2000000	0.4988 (0.0027)

TABLE 5. Computational time for optimization of LHDs with different sizes using different heuristic algorithms.

Size of LHD	Algorithms	MESE	ESE	TPMESE	TPESE	ILS	LSGA	PermGA
30×3	Time/s	7	9	**5**	7	24	/	/
40×4		46	59	**37**	47	762	/	/
50×5		52	89	**44**	78	553	/	/
60×6		68	111	**62**	97	291	627	4285
100×10		547	867	**488**	813	921	1946	/

has the same evaluation number of ϕ_p criterion. Moreover, in Tables 4 to 6, the best values gotten in different stages of tests are already marked by bold (the differences between the results marked and others are also statistically significant at the same measurement point).

1) COMPARISON OF DIFFERENT HEURISTIC ALGORITHMS

To validate the improved performance of our proposed algorithms, the TPMESE, MESE, TPESE and ESE algorithms with other 3 famous heuristic algorithms, PermGA, LSGA and ILS, were compared. Table 4 shows the results with respect to the evaluation number of ϕ_p criterion. The following analysis is based on Table 4.

The TPMESE, MESE and each original algorithm, TPESE and ESE, absolutely show better performance than other algorithms for most of tested cases except the tests of 100×10 LHD. This can be supported by not only the smaller mean values of ϕ_p criterion over 100 runs, but also the p-values of "one of TPMESE, MESE, TPESE and ESE-one of ILS, LSGA and PermGA*" in t-test. The worst algorithm is the PermGA. The largest mean values and the p-values of "PermGA-*" can confirm it. The performance of the ILS is closest to ESE algorithm and its variants, but there is still statistically significant

TABLE 6. Numerical results for optimization of LHDs with different sizes using TPMESE, MESE and each original algorithm.

Size of LHD	Algorithms	Set1		Set2		Set3		Global results
		#Evaluations	Mean (Std)	#Evaluations	Mean (Std)	#Evaluations	Mean (Std)	Mean (Std)
30×3	MESE	60000	1.9842 (0.0249)	500000	**1.9350 (0.0081)**	2000000	1.9245 (0.0052)	**1.9193 (0.0097)**
	ESE	60000	1.9924 (0.0244)	500000	**1.9353 (0.0081)**	2000000	**1.9206 (0.0082)**	**1.9176 (0.0098)**
	TPMESE	60000	**1.9811 (0.0230)**	500000	**1.9347 (0.0080)**	2000000	**1.9211 (0.0081)**	**1.9179 (0.0097)**
	TPESE	60000	1.9887 (0.0220)	500000	**1.9342 (0.0097)**	2000000	**1.9209 (0.0097)**	**1.9181 (0.0101)**
40×4	MESE	150000	1.3475 (0.0085)	1000000	1.3174 (0.0065)	2000000	**1.3080 (0.0052)**	**1.3002 (0.0061)**
	ESE	150000	1.3568 (0.0102)	1000000	1.3212 (0.0099)	2000000	**1.3089 (0.0070)**	**1.3001 (0.0061)**
	TPMESE	150000	**1.3438 (0.0088)**	1000000	**1.3153 (0.0063)**	2000000	**1.3083 (0.0047)**	**1.2998 (0.0053)**
	TPESE	150000	1.3484 (0.0088)	1000000	1.3187 (0.0078)	2000000	**1.3084 (0.0061)**	**1.3016 (0.0054)**
50×5	MESE	150000	1.0212 (0.0069)	1000000	**0.9964 (0.0043)**	2000000	**0.9871 (0.0036)**	**0.9774 (0.0047)**
	ESE	150000	1.0285 (0.0075)	1000000	1.0007 (0.0060)	2000000	0.9912 (0.0049)	**0.9777 (0.0049)**
	TPMESE	150000	**1.0191 (0.0053)**	1000000	**0.9961 (0.0041)**	2000000	**0.9881 (0.0043)**	**0.9777 (0.0040)**
	TPESE	150000	1.0262 (0.0059)	1000000	1.0011 (0.0055)	2000000	0.9923 (0.0055)	**0.9790 (0.0048)**
60×6	MESE	150000	0.8201 (0.0039)	2000000	**0.7936 (0.0029)**	4000000	**0.7867 (0.0026)**	**0.7814 (0.0028)**
	ESE	150000	0.8264 (0.0038)	2000000	0.7976 (0.0034)	4000000	0.7907 (0.0027)	0.7824 (0.0034)
	TPMESE	150000	**0.8185 (0.0042)**	2000000	**0.7931 (0.0031)**	4000000	**0.7875 (0.0028)**	**0.7815 (0.0028)**
	TPESE	150000	0.8251 (0.0036)	2000000	0.7964 (0.0033)	4000000	0.7899 (0.0028)	0.7823 (0.0031)
100×10	MESE	1000000	0.4466 (0.0011)	4000000	0.4410 (0.0007)	8000000	0.4378 (0.0006)	**0.4333 (0.0007)**
	ESE	1000000	0.4490 (0.0013)	4000000	0.4414 (0.0008)	8000000	0.4383 (0.0006)	**0.4331 (0.0007)**
	TPMESE	1000000	**0.4459 (0.0008)**	4000000	**0.4407 (0.0007)**	8000000	**0.4374 (0.0005)**	**0.4332 (0.0005)**
	TPESE	1000000	0.4481 (0.0012)	4000000	0.4411 (0.0009)	8000000	0.4380 (0.0006)	**0.4331 (0.0008)**

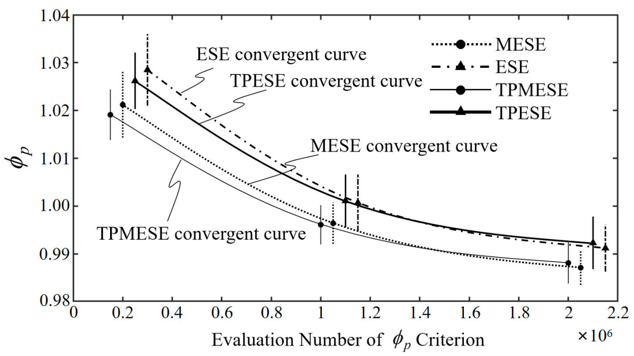

FIGURE 7. Comparison of optimization performance of different algorithms with 50 × 5 LHD.

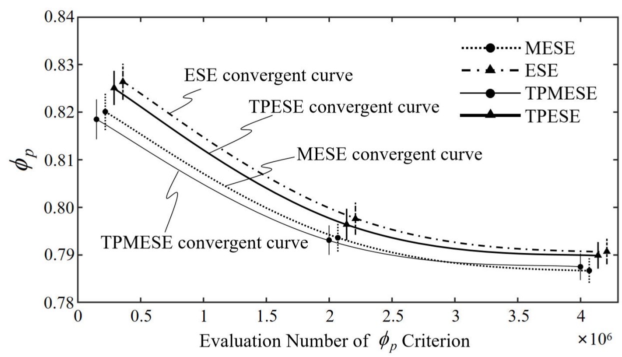

FIGURE 8. Comparison of optimization performance of different algorithms with 60 × 6 LHD.

difference between them in terms of p-values of "ILS-ESE*" for most of tested cases except the case with the largest size. For the tests of 100 × 10 LHD, ILS slightly converges faster than ESE but still is worse than TPMESE and MESE at the first monitored evaluation number of ϕ_p criterion. However, its leading position is quickly reversed by ESE. The smaller mean value of ϕ_p criterion obtained by ESE and the p-values of "ESE-ILS*" at the second monitored evaluation number

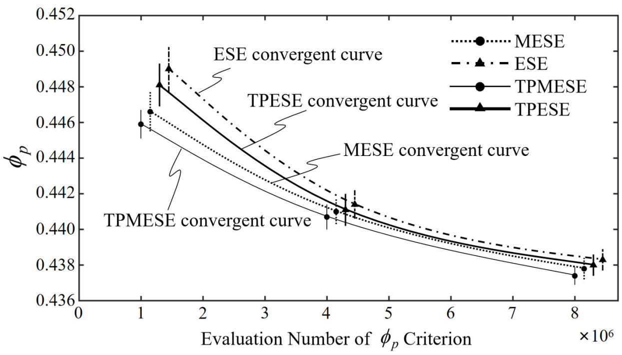

FIGURE 9. **Comparison of optimization performance of different algorithms with 100 × 10 LHD.**

of ϕ_p criterion illustrate it. Therefore, the ILS, LGSA and PermGA algorithms are significantly time-consuming compared with ESE and its variants, if we expect them to converge to a sufficient near-optimal or optimal solution.

In addition, according to the results in Table 4, the performance of MESE and TPMSE are better than each original algorithm, and both converging faster than ESE in the entire corresponding measurement interval except the optimization of LHD with the smallest size. All of smaller mean values respectively obtained by TPMESE and MESE, and the p-values of "TPMESE-MESE*", "MESE-ESE*" and "TPMESE-ESE*" can verify the conclusion, while the p-values of "TPMESE-MESE*", "MESE-ESE*" and "TPMESE-ESE*" are only satisfied at the beginning with respect to the tests of 30 × 3 LHD. Therefore, our proposed algorithms just outperform each original algorithm, TPESE and ESE, at the initial measurement for the optimization of 30 × 3 LHD. In addition, optimization starting from a TPLHD can accelerate the convergence at the beginning, because under the first monitored evaluation number of ϕ_p criterion for all of cases, TPMSE and TPESE can always converge to better designs than MESE and ESE, respectively. The p-values of "TPMESE-MESE*" and "TPESE-ESE*" can confirm it too. However, with the increase of evaluations of ϕ_p criterion, the acceleration of convergence obtained from a TPLHD as an initial design become no longer conspicuous, except the tests of LHD with the largest size. This can also be validated by the p-values of "TPMESE-MESE" and "TPESE-ESE" at the second monitored evaluation number of ϕ_p criterion with respect to the tests of 30 × 3, 40 × 4, 50 × 5 and 60 × 6 LHDs. For the optimization of 100 × 10 LHD, the acceleration originating from a TPLHD as an initial design influences the whole interval of measurement. Correspondingly, the p-values of "TPMESE-MESE*" and "TPESE-ESE*" illustrate it. The reason is simple. On the one hand, the optimization initializing from a TPLHD can directly optimize a near high-quality design almost without time costs, on the other hand, improvement designs can be easily searched owing to high diversity of individuals. These factors significantly promote convergence at the initial stage. As a cogent evidence, optimization can always begin from a TPLHD whose value of ϕ_p criterion is 1.6412 for the tests of 40 × 4 LHD, while the ESE has to evaluate at least several thousand evaluations of ϕ_p criterion to reach the same level solution. However, with the lack of diversity between different designs, it will become more and more difficult to generate a better individual, especially for low-dimensional LHD. Therefore, the leading-position originating from initial design is no longer obvious. In contrast, the algorithm has to take more time escaping from the local solution, which can be reflected by the gradual variations of p-values from "TPMESE-MESE*" and "TPESE-ESE*" to "TPMESE-MESE" and "TPESE-ESE".

Table 5 shows the computational time of different algorithms to reach the same near-optimal value of ϕ_p criterion, where the near-optimal value of ϕ_p criterion for each size test is lower than 104% the final mean value of ϕ_p criterion listed in Table 6. To ensure the statistical significance of the time difference between different algorithms, the results over 100 time runs were averaged on the one hand, on the other hand, t-test was used to compare mean values of computational time for all tests. Meanwhile, the standard of p-value was set to 0.05%. Additionally, 5, 8, and 10 million evaluation numbers of ϕ_p criterion are considered as the thresholds for the tests of LHDs with small, medium and large size, respectively. If no corresponding near-optimal solution was obtained within the evaluation number, optimization will be treated to divergence. We used the symbol "/" to indicate algorithm that can not converge to our near-optimal solution over half of 100 runs in Table 5.

According to the results shown in Table 5, our proposed algorithms, TPMESE and MESE, even each of original algorithms, TPESE and ESE are significantly efficient than other three heuristic algorithms, which could be confirmed by the p-values of "one of TPMESE, MESE, TPESE and ESE-one of ILS, LSGA and PermGA*". PermGA and LSGA show poor global exploration capability to search for a near-optimal solution, especially for the cases of LHDs with small and medium sizes. The TPMESE and MESE are faster than each original algorithm, TPESE and ESE, which can also be confirmed by the p-values of "TPMESE-TPESE*" and "MESE-ESE*". In general, ESE is the most inefficient algorithm in the four approximate algorithms, namely TPMESE, MESE, TPESE and ESE, and it is still significantly faster than ILS, LSGA and PermGA. Correspondingly, the p-values of "TPMESE-ESE*", "MESE-ESE*", "TPESE-ESE*" and "ESE-one of ILS, LSGA and PermGA*" validate statistically significant differences between them.

2) INTERNAL COMPETITION OF PROPOSED ALGORITHMS AND EACH ORIGINAL ALGORITHM

In this part, we compared TPMESE and MESE with each of original algorithms, TPESE and ESE, in the whole period of convergence. The reason why we only compared the four approximate algorithms was that ILS, LSGA and PermGA are significantly time-consuming to reach a near-optimal solution of the same level as ESE and its variants, which we already discussed in above part.

There are different tendencies of performance for optimization with two types of small size LHDs by using different algorithms. TPMESE and MESE can acquire a better design slightly faster than each original algorithm, TPESE and ESE. They also both converge faster than ESE at the initial stage of optimization. The smaller mean values of ϕ_p criterion and the p-values of "TPMESE-TPESE*", "MESE-ESE*" and "TPMESE-ESE*" can both validate the conclusion. However, MESE is quickly exceeded by other algorithms. With the increase of evaluation number, the convergent speed and mean values of the ϕ_p criterion for TPMESE, TPESE, and ESE are almost close to each other but gradually better than MESE. This conclusion can be validated by the p-values based on the results at the 0.5 and 2 million evaluation numbers of ϕ_p criterion. All of the p-values between two random sets of data obtained from different algorithms are larger than standard at the 0.5 million evaluation number on the one hand, on the other hand, the p-values of "MESE-*" at the 2 million evaluation number can confirm the point. For the final results after reaching global stopping criterion, four algorithms can converge to four approximate criterion values. The p-values are all lager than standard, which can confirm this analysis. In addition, the optimization starting from a TPLHD can accelerate convergence at the beginning. The smaller mean values of ϕ_p criterion respectively obtained by TPMESE and TPESE compared with MESE and ESE, as well as the p-values of "TPMESE-MESE*" and "TPESE-ESE*" at the 0.5 evaluation number of ϕ_p criterion can confirm this point.

The optimization with 40 × 4 LHD, TPMESE and MESE show better performance than each of original algorithms, TPESE and ESE, and both converge faster than ESE within 1 million evaluations of ϕ_p criterion. Here, TPMESE and MESE have smaller mean values than each original algorithm on the one hand, on the other hand, the p-values of "TPMESE-TPESE*", "MESE-ESE*" and "TPMESE-ESE*" denote difference is statistical significance. With the increase of evaluations of ϕ_p criterion, it can be predicted that the tendency of convergence above described will disappear, which can be reflected from their close mean values of ϕ_p criterion and the large p-values between different algorithms. All p-values were both gradually larger than our standard after 1 million evaluation number of ϕ_p criterion

In other words, four algorithms converge to four approximate optimal solutions with similar speed. Similarly, the TPLHD as an initial design can also promote faster convergence at the beginning, which can be confirmed by the smaller mean values of ϕ_p criterion respectively obtained by TPMESE and TPESE, as well as the p-values of "TPMESE-MESE∗" and "TPESE-ESE∗" at the first monitored evaluation number of ϕ_p criterion.

Regarding the optimization of LHDs with medium and large sizes, the performance of the TPMESE and MESE algorithms are generally better than TPESE and ESE except the tests of 100 × 10 LHD until the global convergence is reached. This conclusion can be drawn from the results of Table 6 and Figures 7 to 8. For the tests of 50 × 5 and 60 × 6 LHDs, TPMESE and MESE can always converge to a better solution faster than ESE and its variant, while the smaller mean values respectively obtained by TPMESE and MESE as well as all p-values of "MESE-ESE*", "MESE-TPESE*", "TPMESE-ESE*" and "TPMESE-TPESE*" at the all monitored evaluation numbers of ϕ_p criterion confirm it. Finally, these algorithms reach the same level solutions, which can be confirmed by the p-values that are both larger than standard for tests of 50 × 5 LHD, while TPMESE and MESE are still better than TPESE and ESE for the tests of 60 × 6 LHD. In addition, TPMESE and TPESE also respectively perform better than MESE and ESE algorithms at the first monitored evaluation number of ϕ_p criterion. As they can quicker reach a smaller mean value of ϕ_p criterion on the one hand, on the other hand, their mean values of ϕ_p criterion are statistically significant difference from each other in accordance with the p-values of "TPMESE-MESE*" and "TPESE-ESE*".

For the tests of 100 × 10 LHD, TPMESE and MESE show better performance than each of original algorithms, TPESE and ESE, respectively. A conclusion can be drawn based on not only the smaller mean values of ϕ_p criterion respectively obtained by TPMESE and MESE, but also the p-values of "TPMESE-TPESE*" and "MESE-ESE*". Of course, TPMESE and MESE both converge faster than ESE algorithm, until reaching the global convergence. The p-values of "TPMESE-ESE*" and "MESE-ESE*" can confirm it too. Different from the conclusion of other tested cases, TPLHD as an initial design can always promote a faster convergence of optimization than the optimization starting from a random LHD owing to high diversity of individuals. Form the results in Table 6 and Figure 9, TPMESE and TPESE can always search a better solution than MESE and ESE, respectively, before reaching the global convergence. The p-values of "TPMESE-MESE*" and "TPESE-ESE*" can also support the point.

D. TIME SAVINGS OF NEW ALGORITHMS

According to the discussion in Subset C, TPMESE and MESE are more efficient than TPESE and ESE in the whole period of convergence, respectively, particularly at the beginning stage of optimization, except in the tests of small size LHDs. Because of the considerable time costs of the optimization for LHDs with large size, optimizing efficiency has become a top priority. Here, we focus on how efficiency of the TPMESE and MESE algorithms can be improved compared to that of TPESE and ESE algorithms for optimization of large size LHDs. To expound the total time savings, mean time costs for optimization of 60 × 6 and 100 × 10 LHDs using local and global stopping criteria are listed in Table 7. It should be noted that only a near-optimal or sufficient near-optimal design needs to be quickly achieved sometime, particularly when the optimization of LHD with large size is performed.

Because we have already discussed the computational time for generating a near-optimal design in Subset C, a reasonable local stopping criterion for searching a sufficient near-optimal design is set to abort the computation when the value of ϕ_p criterion is lower than 102% of the final mean value shown in Table 6. The entire process was also repeated 100 times to obtain average results of time costs. All results were validated by t-test to ensure there is statistically significant difference from each other. The p-values are both smaller than our standard, where the standard is 0.05%, except the tests of 60 × 6 LHD. The p-values of TPMESE-MESE and TPESE-ESE indicate the acceleration from a TPLHD disappears for the tests of 60 × 6 LHD. Due to different performance of computers provides different results of time costs, the time ratio, which indicates time costs of MESE, TPESE, and ESE normalized by results of MESE, is proposed to illustrate specific time savings.

TABLE 7. Comparison of time costs for different algorithms.

Size	Type	Local time ratio	Global time ratio
60×6	TPMESE/MESE	0.9724	1.0221
	TPESE/MESE	1.3734	1.4564
	ESE/MESE	1.4908	1.4309
100×10	TPMESE/MESE	0.8294	0.9472
	TPESE/MESE	1.2862	1.5016
	ESE/MESE	1.3256	1.5625

From the results listed in Table 7, the TPMESE algorithm has the fastest convergence compared to the other algorithms considered using a local stopping criterion; the maximum improvement is respectively more than 50% and 45% for the tests of 60 × 6 and 100 × 10 LHDs optimized by the ESE algorithm. In the next stage, the acceleration from a TPLHD as an initial design disappears under the global stopping criterion for the tests of 60 × 6 LHD, which can be confirmed by the p-values of "TPMESE-MESE" and "TPESE-ESE" after reaching the global convergence, while the TPMESE algorithm still continues to be the most efficient algorithm for 100 × 10 LHD. The largest time savings exceeded 45% and 60% for tests of 60 × 6 and 100 × 100 LHDs, respectively. Evidently, enhancement of efficiency for global convergence of the new algorithms tends to increase with enlargement of design size. In general, the MESE algorithm converges faster than the ESE algorithm for optimization with large size LHDs. Meanwhile, optimization initializing from a TPLHD is suitable for further enhancing convergence in the initial stage. Therefore, for the optimization of large size LHDs, the TPMESE algorithm is best suited for obtaining a sufficient near-optimal solution. The two types of new algorithms have a few differences in efficiency for tests with different size LHDs under global convergence. It is reasonable to select TPMESE or MESE algorithms to implement global optimization in real-world situations.

E. ANALYSIS OF PROPERTIES FOR MESE ALGORITHM IN OPTIMIZATION

From the discussion in Subsets B and C, the MESE algorithm shows better performance in the global optimization of medium and large size LHDs, and its efficiency is considerably higher than that of the ESE algorithm. To further investigate the reason behind its better performance, we compared the properties of convergence, "temperature" T_h, acceptance number n_{acpt}, and improvement number n_{imp} changed with generation for MESE and ESE algorithms started from a random LHD. To enable the results to be universal, we also averaged the results of optimization for 50 × 5 LHD that was repeated 100 times. However, as the efficiency of optimization for medium and large size LHDs is our main concern, only the results within 200 generations are exhibited. The MESE and ESE algorithms only present a difference in the update method of "temperature"; therefore, the number of evaluations of ϕ_p criterion per generation is the same. Hence, the properties changed with generation can also be equivalent to the shift with evaluation number of ϕ_p criterion.

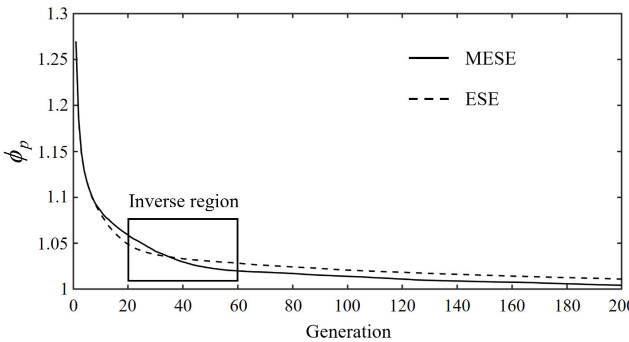

FIGURE 10. Comparison of convergence properties for MESE and ESE algorithms. The inverse region indicates an interval where the convergence of the ESE algorithm is exceeded by the MESE algorithm. Then, the MESE algorithm converges faster than the ESE algorithm with generation.

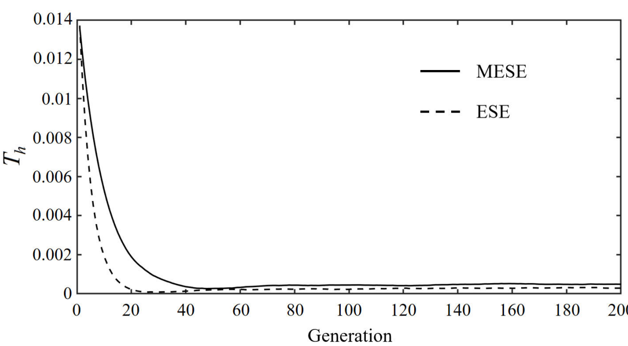

FIGURE 11. Comparison of "temperature" shifted with generation for MESE and ESE algorithms.

From Figures 10 and 11, we can see that a slightly slow reduction of "temperature" in the initial stage and smoother fluctuation in the following generations produce faster convergence. The reason that leads to different properties of

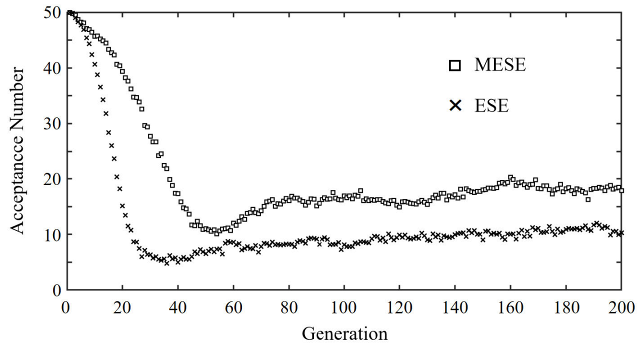

FIGURE 12. Comparison of acceptance number shifted with generation for MESE and ESE algorithms.

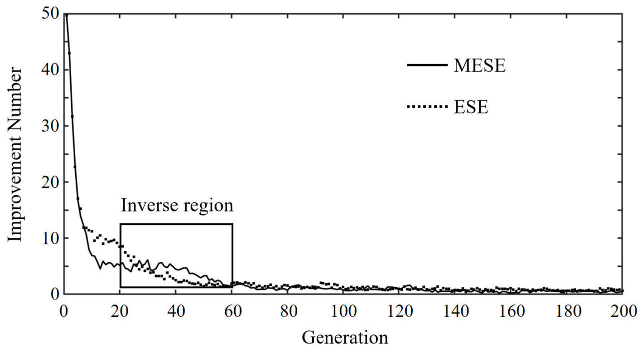

FIGURE 13. Comparison of improvement number shifted with generation for MESE and ESE algorithms. The inverse region indicates an interval where the number of improvement designs generated by the ESE algorithm is surpassed by the MESE algorithm.

"temperature" shifted with generation for ESE and MESE algorithms is that there is a considerable diversity of individuals at the optimization beginning. Thus, every generation can produce abundant acceptance and improvement designs, which can be reflected through large acceptance n_{acpt} and improvement numbers n_{imp} in Figures 12 and 13. Therefore, at this stage, the ESE algorithm always performs an improvement process to quickly search local solutions by rapidly reducing T_h in terms of expression $T_h = \alpha_1 T_{h\_old}$, while the $T_h$of MESE is only quickly decreased, when the acceptance ratio n_{acpt}/M is close to 1. With the reduction of acceptance number n_{acpt}, the T_h of MESE tends to slower reduction using a larger scale factor in terms of Eq. (6). Correspondingly, according to Figures 11 and 12, there is no significant difference of T_h shifted between MESE and ESE with relatively larger acceptance number within 5 generations, such as the n_{acpt} which is larger than 45. However, when the n_{acpt} is lower than 45 after 5 generations, the two variation curves of the T_h begin to show different tendencies. The T_h of ESE continues to be rapidly reduced to a considerably small value which is close to 0, while the T_h of MESE maintains a smoother reduction. Accordingly, rapid reduction of T_h for ESE also implies rapid decrease of diversity, which leads to a fast loss of necessary intermediate solutions in accordance with considerably smaller acceptance number n_{acpt} of ESE within 20 to 40 generations in Figure 12. As a consequence, ESE no longer has enough intermediate solutions to generate abundant improvement designs. Therefore, the improvement number n_{imp} and corresponding value of ϕ_p criterion are fast reversed by MESE within the same interval, 30 to 40 generations as is mentioned above. The inverse regions in Figures 10 and 13 confirm the analysis. Even if ESE can increase T_h to escape from a local solution after deficiency of diversity begun from 35 generations from Figure 11, it is still difficult to rapidly recover necessary acceptance number so that the algorithm can continue to generate a lot of improvement designs. Thus, the MESE always keeps leading position compared to ESE algorithm in the following generations. In contrast, for the optimization of LHDs with small size, in the later period of convergence, the larger T_h and the corresponding acceptance number with respect to the TPMESE and MESE algorithms imply that there are still a lot of intermediate solutions. In other words, a number of replacements between the current design X and the intermediate design X_{try} are still conducted in the inner loops of MESE and TPMESE. However, due to the low diversity among designs in the optimization of LHDs with small size, and the continuous replacements (substantially mutations), the relatively high mutations among designs in terms of a number of replacements between X and X_{try} are not beneficial to remain the current superior design X so that the convergent speed is slowed down in the later period of convergence. It illustrates the reason why the leading positing of TPMESE and MESE disappear quickly compared to ESE algorithm along with the increase of evaluations of ϕ_p criterion, as is analyzed in Section C, for the optimization of LHDs with small size.

In general, the property of update "temperature" for the MESE algorithm is more suitable for the optimization of medium and large size LHDs than the ESE algorithm. That is the main reason why the efficiency of the MESE algorithm is higher than that of the ESE algorithm in these situations.

V. CONCLUSION

In this study, a more efficient algorithm, MESE, than prevalent algorithms is proposed to optimize LHD. This algorithm is modified from the ESE algorithm by introducing a new update method of "temperature", whose shift is controlled by scale factors of variable step length. Furthermore, considering the unique advantage of TPLHD which needs no computational process to rapidly construct a near high-quality design, we combine the new MESE algorithm with TPLHD (TPMESE) to further improve its performance in the beginning of optimization. The performance of the new algorithms is verified by optimization tests on five sizes of LHDs, classified into three levels: small, medium, and large. From a comparison of the results, the following conclusions are obtained:

1)The new algorithms, TPMESE and MESE, are considerably more efficient than ILS, LSGA and PermGA for all of tested cases, and more suitable for the exploration of a more optimal solution than LSGA and PermGA.

2)The TPMESE and MESE algorithms show significant improvement in efficiency compared to that of each original algorithm, TPESE and ESE, and they converge faster than ESE, in the optimizations for medium and large size LHDs, while they only perform better than each original algorithm at the beginning of optimization for the tests of small size LHDs. Finally, The four algorithms, TPMESE, MESE, TPESE and ESE, exhibit approximate performance of global exploration with each other to search for an optimal solution.

3)The optimization starting from a TPLHD effectively promotes convergence at the beginning, while this effect gradually disappears in the subsequent simulation. Therefore, if there are no time constraints, MESE or TPMESE algorithms are good choices to quickly reach an optimal solution. However, when time cost is our top priority, especially for medium or large size LHD, the TPMESE algorithm is better than the MESE algorithm for generating a near-optimal or sufficient near-optimal solution after several million evaluations of ϕ_p criterion.

4) The proposed method to update the ''temperature'' in the MESE algorithm has different properties from the ESE algorithm. The updating process in the MESE algorithm is suitable to balance capabilities of local exploitation and global exploration of the algorithm in the optimizations of medium and large size LHDs. Therefore, the proposed algorithms can converge faster to obtain an optimal solution.

It is worth noting that even though our proposed algorithms, TPMSE and MESE, are efficient, and have the ability to search for an optimal solution, there are still some limitations in its application. MESE algorithm and its variant have to set many parameters which considerably affect the performance. In addition, the performance of the proposed algorithms is only verified based on the specific computational experiments when the dimension is not greater than 10. They are also worthy of testing for LHDs with higher dimensions and further investigation when proposed algorithms are used in practical applications.

REFERENCES

[1] A. Forrester, A. Sobester, and A. Keane, *Engineering Design Via Surrogate Modelling: A Practical Guide*. Hoboken, NJ, USA: Wiley, 2008, pp. 1–18.

[2] W. J. Roux, N. Stander, and R. T. Haftka, ''Response surface approximations for structural optimization,'' *Int. J. Numer. Methods Eng.*, vol. 42, no. 3, pp. 517–534, Jun. 1998.

[3] M. D. McKay, R. J. Beckman, and W. J. Conover, ''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, May 1979.

[4] R. L. Iman and W. J. Conover, ''Small sample sensitivity analysis techniques for computer models.With an application to risk assessment,'' *Commun. Statist.-Theory Methods*, vol. 9, no. 17, pp. 1749–1842, Jan. 1980.

[5] A. Grosso, A. R. M. J. U. Jamali, and M. Locatelli, ''Finding maximin latin hypercube designs by iterated local search heuristics,'' *Eur. J. Oper. Res.*, vol. 197, no. 2, pp. 541–547, Sep. 2009.

[6] K. Q. Ye, W. Li, and A. Sudjianto, ''Algorithmic construction of optimal symmetric latin hypercube designs,'' *J. Stat. Planning Inference*, vol. 90, no. 1, pp. 145–159, Sep. 2000.

[7] W. W. Li and C. F. Jeff Wu, ''Columnwise-pairwise algorithms with applications to the construction of supersaturated designs,'' *Technometrics*, vol. 39, no. 2, pp. 171–179, May 1997.

[8] K.-T. Fang, C.-X. Ma, and P. Winker, ''Centered L_2-discrepancy of random sampling and Latin hypercube design, and construction of uniform designs,'' *Math. Comput.*, vol. 71, no. 237, pp. 275–296, 2002.

[9] S. Bates, J. Sienz, and V. Toropov, ''Formulation of the optimal latin hypercube design of experiments using a permutation genetic algorithm,'' in *Proc. 45th AIAA/ASME/ASCE/AHS/ASC Struct., Struct. Dyn. Mater. Conf.*, Apr. 2004, pp. 1–7.

[10] R. Jin, W. Chen, and A. Sudjianto, ''An efficient algorithm for constructing optimal design of computer experiments,'' *J. Stat. Planning Inference*, vol. 134, no. 1, pp. 268–287, Sep. 2005.

[11] T. Chantarawong, J. Rungrattanaubol, and A. Na-Udom, ''Enhancement of enhanced stochastic evolutionary algorithm for computer simulated experiment,'' *Inf. Technol. J.*, vol. 6, no. 2, pp. 65–69, 2010.

[12] H. Zhu, L. Liu, T. Long, and L. Peng, ''A novel algorithm of maximin latin hypercube design using successive local enumeration,'' *Eng. Optim.*, vol. 44, no. 5, pp. 551–564, May 2012.

[13] R.-B. Chen, D.-N. Hsieh, Y. Hung, and W. Wang, ''Optimizing latin hypercube designs by particle swarm,'' *Statist. Comput.*, vol. 23, no. 5, pp. 663–676, Sep. 2013.

[14] N. Pholdee and S. Bureerat, ''An efficient optimum latin hypercube sampling technique based on sequencing optimisation using simulated annealing,'' *Int. J. Syst. Sci.*, vol. 46, no. 10, pp. 1780–1789, Jul. 2015.

[15] M. R. Kianifar, F. Campean, and A. Wood, ''PermGA algorithm for a sequential optimal space filling DoE framework,'' in *Proc. 14th UK Workshop Comput. Intell. (UKCI)*, Sep. 2014, pp. 1–8.

[16] H. Mahmoudi and H. Zimmermann, ''On optimal latin hypercube design for yield analysis of analog circuits,'' in *Proc. Austrian Workshop Microelectron.*, Vienna, Austria, Sep. 2015, pp. 46–49.

[17] S. Ba, W. R. Myers, and W. A. Brenneman, ''Optimal sliced latin hypercube designs,'' *Technometrics*, vol. 57, no. 4, pp. 479–487, Nov. 2015.

[18] V. R. Joseph, E. Gul, and S. Ba, ''Maximum projection designs for computer experiments,'' *Biometrika*, vol. 102, no. 2, pp. 371–380, Jun. 2015.

[19] V. R. Joseph, ''Space-filling designs for computer experiments: A review,'' *Qual. Eng.*, vol. 28, no. 1, pp. 28–35, Jan. 2016.

[20] T. Long, D. Wu, X. Chen, X. Guo, and L. Liu, ''A deterministic sequential maximin latin hypercube design method using successive local enumeration for metamodel-based optimization,'' *Eng. Optim.*, vol. 48, no. 6, pp. 1019–1036, Jun. 2016.

[21] K. Le Guiban, A. Rimmel, M.-A. Weisser, and J. Tomasik, ''The first approximation algorithm for the maximin latin hypercube design problem,'' *Oper. Res.*, vol. 66, no. 1, pp. 253–266, Feb. 2018.

[22] Z. Wu, D. Wang, P. N. Okolo, K. Zhao, and W. Zhang, ''Efficient space-filling and near-orthogonality sequential latin hypercube for computer experiments,'' *Comput. Methods Appl. Mech. Eng.*, vol. 324, pp. 348–365, Sep. 2017.

[23] W. Li, L. Lu, X. Xie, and M. Yang, ''A novel extension algorithm for optimized latin hypercube sampling,'' *J. Stat. Comput. Simul.*, vol. 87, no. 13, pp. 2549–2559, Jun. 2017.

[24] Q. Xiao and H. Xu, ''Construction of maximin distance latin squares and related latin hypercube designs,'' *Biometrika*, vol. 104, no. 2, pp. 455–464, Feb. 2017.

[25] X. Shang, T. Chao, P. Ma, and M. Yang, ''An efficient local search-based genetic algorithm for constructing optimal latin hypercube design,'' *Eng. Optim.*, vol. 52, no. 2, pp. 271–287, Feb. 2020.

[26] F. A. C. Viana, G. Venter, and V. Balabanov, ''An algorithm for fast optimal latin hypercube design of experiments,'' *Int. J. Numer. Methods Eng.*, vol. 82, no. 2, pp. 135–156, Apr. 2010.

[27] G. Damblin, M. Couplet, and B. Iooss, ''Numerical studies of space-filling designs: Optimization of latin hypercube samples and subprojection properties,'' *J. Simul.*, vol. 7, no. 4, pp. 276–289, Nov. 2013.

[28] B. G. M. Husslage, G. Rennen, E. R. van Dam, and D. den Hertog, ''Space-filling latin hypercube designs for computer experiments,'' *Optim. Eng.*, vol. 12, no. 4, pp. 611–630, Dec. 2011.

[29] M. D. Morris and T. J. Mitchell, ''Exploratory designs for computational experiments,'' *J. Stat. Planning Inference*, vol. 43, no. 3, pp. 381–402, Feb. 1995.

[30] Y. G. Saab and V. B. Rao, ''Combinatorial optimization by stochastic evolution,'' *IEEE Trans. Comput.-Aided Design Integr.*, vol. 10, no. 4, pp. 525–535, Apr. 1991.

[31] A. W. Mohamed and H. Z. Sabry, ''Constrained optimization based on modified differential evolution algorithm,'' *Inf. Sci.*, vol. 194, pp. 171–208, Jul. 2012.

QINGYU WANG is currently pursuing the Ph.D. degree in fluid dynamics for transportation and environmental systems with the School of Engineering, Hiroshima University, Japan. He is also supported by the China Scholarship Council to focus on the establishment of a new optimization system for vehicle aerodynamics. His main research interests include optimization of vehicle aerodynamics and investigation of application of evolutionary algorithms in optimization.

TAKUJI NAKASHIMA was born in Osaka, Japan, in 1977. He received the Doctor of Engineering degree from The University of Tokyo, in 2006. He is currently an Assistant Professor with the Graduate School of Advanced Science and Engineering, Hiroshima University, Japan. He has a strong interest in data science and applied mathematics regarding the CFD with high-performance computing technology, such as algorithms for optimization and graph partitioning. His main research interests include applied aerodynamics using both computational fluid dynamics (CFD), experimental fluid dynamics (EFD) in automobile, and ship building engineering.

CHENGUANG LAI received the Ph.D. degree from Tohoku University. He is currently a Professor with the College of Vehicle Engineering, Chongqing University of Technology, China. He is also a Collaborative Researcher and a Visiting Professor with Tohoku University. His main research interests include intelligent optimization and analysis for vehicle aerodynamics, vehicle body design, and body cooperative optimization.

HIDEMI MUTSUDA was a Visiting Researcher with Stanford University, the Norwegian University of Science and Technology, and the Japan Agency for Marine-Earth Science and Technology. He is currently a Professor with the Graduate School of Advanced Science and Engineering, Hiroshima University, Japan. His main research interests include environmental fluid engineering and computational fluid dynamics for vehicle, ocean/atmosphere, and energy harvesting.

TAIGA KANEHIRA received the Doctor of Engineering degree from Hiroshima University, Japan, in 2020. From 2019 to 2020, he was a JSPS Research Fellowship for Young Scientists. He is currently an Assistant Professor with the Graduate School of Advanced Science and Engineering, Hiroshima University. His research interests include computational fluid dynamics and particle-based method for vehicle, ocean structures, and ocean waves.

YASUFUMI KONISHI was born in Osaka, Japan, in 1974. He received the Doctor of Engineering degree from the Tokyo Metropolitan Institute of Technology, in 2006. He has been a Senior Fellow with the Advanced Flow Experimental Research Center, Institute of Fluid Science, Tohoku University, since 2019. His current research interests include advanced wind tunnel testing techniques, flow instability, and aerodynamics of sport equipment.

HIROYUKI OKUIZUMI was born in 1983. He received the Bachelor of Science degree from Niigata University, in 2006. He has been a Technical Staff with the Advanced Flow Experimental Research Center, Institute of Fluid Science, Tohoku University, Japan. His research interests include operation, maintenance, management, and testing assistance of wind tunnel.