

A Two-Phase Differential Evolution for Uniform Designs in Constrained Experimental Domains

Yong Wang, *Member, IEEE*, Biao Xu, Guangyong Sun, and Shengxiang Yang, *Senior Member, IEEE*

Abstract—In many real-world engineering applications, a uniform design needs to be conducted in a constrained experimental domain that includes linear/nonlinear and inequality/equality constraints. In general, these constraints make the constrained experimental domain small and irregular in the decision space. Therefore, it is difficult for current methods to produce a predefined number of samples and make the samples distribute uniformly in the constrained experimental domain. This paper presents a two-phase differential evolution for uniform designs in constrained experimental domains. In the first phase, considering the constraint violation as the fitness function, a clustering DE is proposed to guide the population toward the constrained experimental domain from different directions promptly. As a result, a predefined number of samples can be obtained in the constrained experimental domain. In the second phase, maximizing the minimum Euclidean distance among samples is treated as another fitness function. By optimizing this fitness function, the samples produced in the first phase can be scattered uniformly in the constrained experimental domain. The performance of the proposed method has been tested and compared with another state-of-the-art method. Experimental results suggest that our method is significantly better than the compared method in the uniform designs of a new type of automotive crash box and five benchmark test problems. Moreover, the proposed method could be considered as a general and promising

framework for other uniform designs in constrained experimental domains.

Index Terms—A new type of automotive crash box, constrained experimental domain, differential evolution, uniform design.

I. INTRODUCTION

ENTERPRISES always try to provide high-quality, low-cost products and shorten lead time to survive in fierce competitive environments. This can be achieved through innovative design. Traditional product design is based on empirical and/or trial-and-error processes. Utilizing traditional methods to develop a new product will inevitably lead to a long design cycle and excessive costs because of numerous prototype tests. With the rapid development of advanced computer technology, finite element analysis has become a well-established numerical simulation tool to precisely predict product performance in the design stage, which can reduce design and prototyping costs [1]. Since the design scheme should be manually adjusted in line with engineers' experience, finite element analysis needs to be performed many times to reevaluate the results. Furthermore, the above process cannot guarantee the global optimum. In order to achieve more active design, finite element analysis-based optimization has been developed as a promising manner to systematically seek an optimal design [2]. Although finite element analysis-based optimization is able to improve the design level, it is essentially an iterative process. Note that computation-intensive design problems are becoming increasingly common in automotive, aerospace, transportation, and defense industries [3]. With respect to such design problems, it is time-consuming for one iteration. For example, in full vehicle crashworthiness, to meet all safety criteria, the design is intrinsically a daunting optimization task often involving multiple loading cases (e.g., frontal/side/rear impact, pedestrian safety, roof crush, interior head impact, and rollover), multiple disciplines, and multiple objectives [4]–[6]. Just taking the computational cost into account, it has been reported that it takes Ford Motor Company about 36–160 h to run one full vehicle crash simulation [7]. In general, a full vehicle crashworthiness design needs to call hundreds of simulations. In order to enhance the computational efficiency in design, highly accurate surrogate models have been widely used [8]–[10]. How to generate representative samples is the primary issue for constructing highly accurate surrogate models. The design of experiment can be considered to be an essential step toward this purpose [11].

Manuscript received July 4, 2016; revised November 4, 2016 and February 2, 2017; accepted February 3, 2017. Date of publication February 17, 2017; date of current version September 28, 2017. This work was supported in part by the Innovation-Driven Plan in Central South University under Grant 2015CX012 and Grant 2015CX007, in part by the National Natural Science Foundation of China under Grant 61673397, Grant 61273314, and Grant 61673331, in part by the EU Horizon 2020 Marie Skłodowska-Curie Individual Fellowships under Project 661327, in part by the Engineering and Physical Sciences Research Council of U.K. under Grant EP/K001310/1, in part by the Hunan Provincial Natural Science Fund for Distinguished Young Scholars under Grant 2016JJ1018, in part by the Program for New Century Excellent Talents in University under Grant NCET-13-0596, and in part by the Science Fund of State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body under Grant 31515004. (*Corresponding author: Guangyong Sun*).

Y. Wang is with the School of Information Science and Engineering, Central South University, Changsha 410083, China, and also with the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K. (e-mail: ywang@csu.edu.cn).

B. Xu is with the School of Information Science and Engineering, Central South University, Changsha 410083, China (e-mail: xubiao1189@163.com).

G. Sun is with the School of Aerospace, Mechanical and Mechatronic Engineering, Faculty of Engineering, University of Sydney, Sydney, NSW 2006, Australia, and also with the State Key Laboratory of Advanced Design and Manufacture for Vehicle Body, Hunan University, Changsha 410082, China (e-mail: guangyong.sun@sydney.edu.au).

S. Yang is with the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K. (e-mail: syang@dmu.ac.uk).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TEVC.2017.2669098

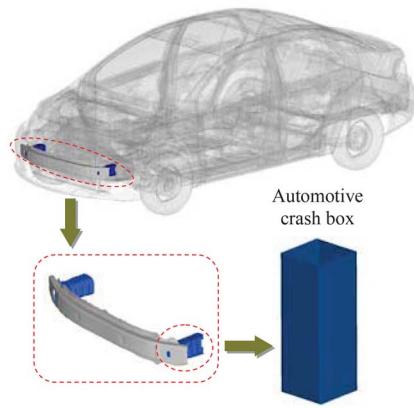


Fig. 1. Illustration of the automotive crash box.

The design of experiment aims to appropriately select a set of samples to provide uniform coverage over the experimental domain of interest [12]. Currently, the design of experiment has been broadly applied to computer experiments and industrial experiments [13]–[15]. Many techniques have been proposed for the design of experiment, such as orthogonal design [16]–[18], Latin square design [19]–[21], and uniform design [22]–[24]. Among them, uniform design is able to obtain more information from the experimental domain with a smaller number of samples [25]. Very often, for uniform designs of some complex practical situations, there are many linear/nonlinear and inequality/equality constraints due to the interactions among the decision variables [26]. The uniform design of a new type of automotive crash box for lightweight can be taken as an example. The automotive crash box (as shown in Fig. 1) always bears very complex loading, which implies that different regions should have different roles to maximize the usage of materials. How to exhaust the potential of materials according to the performance requirements is one of the most important issues for automotive lightweight design. Fortunately, variable-thickness rolled blank can vary the blank thickness with a continuous thickness transition through adjusting the roll gap [27]. This innovative technology can easily realize the customized thickness of a new type of automotive crash box. Note that different thickness distributions of the variable-thickness rolled blanks lead to different performance; therefore, it is necessary to optimize the thickness distributions to maximize performance [28]. The optimal thickness distribution among different zones including the thin zones, thickness zones, and transition zones should satisfy a variety of manufacturing constraints [27], which are typical linear/nonlinear and inequality/equality constraints.

In this paper, the aim of a uniform design is to sample uniformly over a constrained experimental domain which is usually a small and irregular region in the decision space because of constraints. At present, many uniform design approaches have been proposed for this purpose.

1) *Traditional Uniform Design Methods*: This kind of method first generates a number of uniform samples from the decision space, and then checks which samples satisfy all constraints, deleting the samples violating the

constraints [29]–[32]. However, if there exist nonlinear constraints and/or equality constraints, the experimental domain is very small compared to the decision space. As a result, the samples satisfying all constraints are very scarce. Under this condition, it is a very challenging task for traditional uniform design methods to produce a desired number of samples.

2) *Number-Theoretic Uniform Design Methods*: This kind of method generates a number of uniform samples by number-theoretic techniques and tackles constraints by mapping function [33]–[35]. Unfortunately, this kind of method can only handle one linear equality constraint, and the other constraints are handled by the same process as in traditional uniform design methods. Hence, it faces substantial difficulties when dealing with complex nonlinear and/or equality constraints.

3) *Heuristic Uniform Design Methods*: This kind of method models a uniform design in the constrained experimental domain as an optimization problem, and optimizes it via heuristic methods [36]–[39]. However, this kind of method is not suitable for uniform designs in the constrained experimental domains with high dimensionality and a large number of design points due to tremendous computational workload. Additionally, most methods do not introduce how to cope with nonlinear constraints and equality constraints.

From the above introduction, it is obvious that linear/nonlinear and inequality/equality constraints pose a great challenge to current uniform design methods. Therefore, new insights toward uniform designs in constrained experimental domains are quite necessary. In this paper, inspired by [40], a two-phase differential evolution (DE) called ToPDE is proposed. In the first phase, a clustering DE integrated with constraint violation as the fitness function is presented to guide the population (i.e., a set of samples or a set of design points) toward the constrained experimental domain quickly. The advantages of the clustering DE are twofold. On one hand, it is capable of obtaining a predefined number of samples in the constrained experimental domain. On the other hand, it can keep the diversity of the population, with the purpose of providing high-quality candidate solutions for the second phase. In the second phase, another fitness function and a novel replacement strategy are proposed to make the population distribute uniformly in the constrained experimental domain, which can also dramatically reduce the computational time.

The main contributions of this paper can be summarized as follows.

- 1) ToPDE can be viewed as a new algorithmic framework with the incorporation of the properties of uniform designs in constrained experimental domains. Moreover, ToPDE is not dependent on the type and number of constraints.
- 2) Recognizing that the computational overhead of the existing heuristic uniform design methods is very high, the fitness function in the second phase offers a tradeoff between uniform distribution and computational cost.
- 3) ToPDE has been applied to the uniform designs of a practical engineering problem (i.e., a new type of

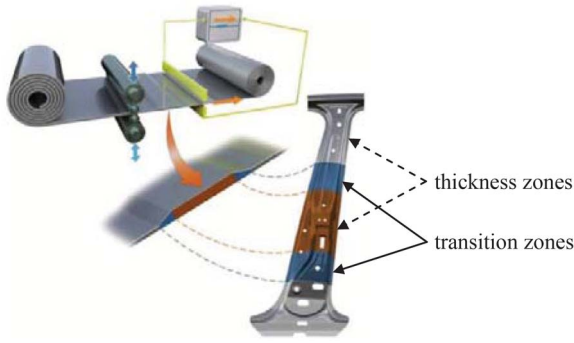


Fig. 2. Rolling processes of the variable-thickness rolled blank [27].

automotive crash box) and five benchmark test problems. The experimental results show that ToPDE is significantly better than NTMD [33], which is a state-of-the-art number-theoretic method for uniform designs in constrained experimental domains.

This paper is organized as follows. Section II describes the uniform design of a new type of automotive crash box. Section III introduces the concepts of DE. ToPDE is proposed in Section IV. Section V presents the experimental results. Section VI further discusses the advantage and effectiveness of ToPDE. Finally, Section VII concludes this paper.

II. UNIFORM DESIGN OF A NEW TYPE OF AUTOMOTIVE CRASH BOX

Reducing the weight of an automobile is an effective method for saving energy and reducing emissions. The most important premise of automotive lightweight design is not to reduce product performance [41], especially safety. In all automotive safety parts, the crash box shown in Fig. 1 is an irreplaceable part [42] whose energy absorption has a significant effect on occupant safety. To improve the energy absorption, the most commonly used way is to increase the wall thickness of the traditional automotive crash box, which inevitably leads to a significant increase in weight. Therefore, it is very challenging to achieve light weight and high crashworthiness simultaneously. In order to address this issue, it is necessary to design a new type of automotive crash box for the purpose of reducing the weight. Herein, the new type of automotive crash box is designed by variable-thickness rolled blank, the rolling processes of which are shown in Fig. 2 [27]. The variable-thickness rolled blank consists of different thickness zones and transition zones, where the transition zones are used to link different thickness zones. As mentioned, directly combining optimization algorithms with finite element analysis will consume a great deal of time to search for the optimal thickness distribution of the new type of automotive crash box. Indeed, highly accurate surrogate models provide an effective way to improve the computational efficiency. From [11], we know that the key issue of constructing highly accurate surrogate models is to obtain representative samples.

The structure of a new type of automotive crash box is shown in Fig. 3, which includes 14 decision variables, i.e., nine different lengths (x_1, x_2, \dots, x_9) and five different thicknesses

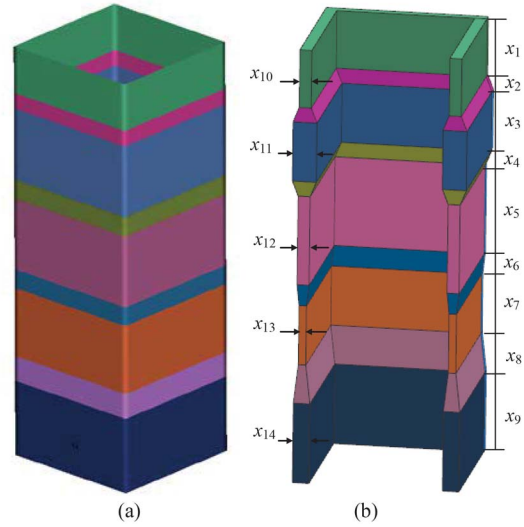


Fig. 3. Structure of a new type of automotive crash box. (a) Overall view of the new type of automotive crash box. (b) Sectional view of the new type of automotive crash box.

TABLE I
RELATIONSHIP OF THE DECISION VARIABLES IN THE NEW TYPE OF AUTOMOTIVE CRASH BOX

Relationship of the decision variables	Type of the constraints
$x_i^{lower} \leq x_i \leq x_i^{upper}, i = 1, 2, \dots, 14$	Boundary constraints
$\sum_{i=1}^9 x_i = L$	Linear equality constraint
$a \leq x_{10} / x_{11} \leq b, a \leq x_{11} / x_{12} \leq b,$ $a \leq x_{12} / x_{13} \leq b, a \leq x_{13} / x_{14} \leq b,$ $c \leq x_{11} - x_{10} / x_2 \leq d, c \leq x_{12} - x_{11} / x_4 \leq d,$ $c \leq x_{13} - x_{12} / x_6 \leq d, c \leq x_{14} - x_{13} / x_8 \leq d,$ $x_1 x_{10} + 0.5 x_2 (x_{10} + x_{11}) + x_3 x_{11} +$ $0.5 x_4 (x_{11} + x_{12}) + x_5 x_{12} + 0.5 x_6 (x_{12} + x_{13}) +$ $x_7 x_{13} + 0.5 x_8 (x_{13} + x_{14}) + x_9 x_{14} \leq \frac{M}{\rho}$	Nonlinear inequality constraints
ρ denotes the density of material	
l denotes the circumference of cross section	

($x_{10}, x_{11}, \dots, x_{14}$). In addition, there are many constraints. First, each decision variable x_i should be in a predefined range, i.e., $[x_i^{lower}, x_i^{upper}]$, $i = 1, 2, \dots, 14$, where x_i^{lower} and x_i^{upper} denote the lower and upper bounds of x_i , respectively. Second, the new type of automotive crash box should satisfy the requirement of the standard industrial size. Therefore, the sum of the length of different thickness zones and transition zones should be equal to a constant L . Third, the transition zones should satisfy the limitations of manufacturing technology, and thus, the ratio of adjacent thicknesses should be in the range of $[a, b]$. Moreover, the ratio between the difference of adjacent thicknesses and the length of the corresponding transition zone should be in the range of $[c, d]$. Finally, the weight of the new type of automotive crash box should be less than or equal to a constant M . The detailed relationships of the decision variables are given in Table I and the parameter values are given in Table II.

TABLE II
PARAMETER VALUES OF THE NEW TYPE OF AUTOMOTIVE CRASH BOX

Parameter	Value	Parameter	Value
$x_1^{lower}, \dots, x_9^{lower}$	1 (mm)	b	1.5
$x_{10}^{lower}, \dots, x_{14}^{lower}$	0.8 (mm)	c	1/30
$x_1^{upper}, \dots, x_9^{upper}$	175 (mm)	d	1/10
$x_{10}^{upper}, \dots, x_{14}^{upper}$	2.5 (mm)	M	2.1168 (Kg)
L	175 (mm)	ρ	2.7×10^{-6} (Kg/mm ³)
a	0.6	l	320 (mm)

From Table I, it can be seen that the experimental domain is subject to 14 boundary constraints, one linear equality constraint, and 17 nonlinear inequality constraints. The ultimate aim of the uniform design of the new type of automotive crash box is to sample uniformly over this constrained experimental domain.

III. CONCEPTS OF DIFFERENTIAL EVOLUTION

DE, proposed by Storn and Price [43], is a population-based optimizer. It is one of the most popular paradigms of evolutionary algorithms and has been successfully applied to solve different kinds of optimization problems [44]–[48]. Suppose that an optimization problem is to minimize the objective function $f(\vec{x})$. First, DE randomly generates NP samples (also called NP individuals) from the decision space

$$\vec{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D}) \in S, i = 1, 2, \dots, NP \quad (1)$$

where

- $x_{i,j}$ ($j \in \{1, 2, \dots, D\}$) j th decision variable;
- D number of decision variables;
- $S = \prod_{j=1}^D [x_j^{lower}, x_j^{upper}]$ decision space;
- x_j^{lower} and x_j^{upper} lower and upper bounds of $x_{i,j}$, respectively.

These NP samples form the initial population, i.e., an $NP \times D$ matrix. Next, DE adopts three main steps, i.e., mutation, crossover, and selection to evolve the population.

A. Mutation

By the mutation, DE creates a mutant vector $\vec{v}_i = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$ for each sample \vec{x}_i . The two extensively used mutation operators (called DE/rand/1 and DE/current-to-rand/1) are introduced as follows.

1) DE/rand/1

$$\vec{v}_i = \vec{x}_{r_1} + F \times (\vec{x}_{r_2} - \vec{x}_{r_3}), \quad i = 1, 2, \dots, NP \quad (2)$$

2) DE/current-to-rand/1

$$\vec{v}_i = \vec{x}_i + rand \times (\vec{x}_{r_1} - \vec{x}_i) + F \times (\vec{x}_{r_2} - \vec{x}_{r_3}), \quad i = 1, 2, \dots, NP \quad (3)$$

where r_1 , r_2 , and r_3 are three random and mutually different integers chosen from $[1, NP]$, $rand$ is a uniformly distributed random number from $[0, 1]$, and $F \in (0, 1]$ is the scaling factor which controls the amplification of the differential vector $(\vec{x}_{r_2} - \vec{x}_{r_3})$.

B. Crossover

The crossover is implemented on \vec{x}_i and its mutant vector \vec{v}_i to produce a trial vector $\vec{u}_i = (u_{i,1}, u_{i,2}, \dots, u_{i,D})$

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } rand_j < CR \text{ or } j = j_{rand} \\ x_{i,j}, & \text{otherwise} \end{cases}, \quad \begin{matrix} i = 1, 2, \dots, NP \\ j = 1, 2, \dots, D \end{matrix} \quad (4)$$

where $CR \in [0, 1]$ is the crossover control parameter, $rand_j$ is the j th evaluation of a uniformly distributed random number between 0 and 1, and j_{rand} is an index randomly chosen from $\{1, 2, \dots, D\}$. The condition “ $j = j_{rand}$ ” ensures that \vec{u}_i differs from \vec{x}_i by at least one element.

C. Selection

\vec{x}_i is replaced by \vec{u}_i if the objective function value of \vec{x}_i is not better than that of \vec{u}_i ; otherwise, \vec{x}_i will survive into the next generation

$$\vec{x}_i = \begin{cases} \vec{u}_i, & \text{if } f(\vec{u}_i) \leq f(\vec{x}_i) \\ \vec{x}_i, & \text{otherwise} \end{cases}, \quad i = 1, 2, \dots, NP. \quad (5)$$

In DE, the mutation, crossover, and selection are executed generation by generation until a predefined stopping criterion is satisfied. From the introduction, it is evident that DE does not include any complex operators.

IV. A TWO-PHASE DE FOR UNIFORM DESIGNS IN CONSTRAINED EXPERIMENTAL DOMAINS

A. Fitness Functions

If a sample satisfies all constraints, then it is called a feasible sample; otherwise, it is called an infeasible sample. A constrained experimental domain is the set of all feasible samples. As pointed out previously, the constrained experimental domain may be a small and irregular region in the decision space because of constraints. The generated samples in the population of DE by the initialization satisfy boundary constraints, but most of them—even all of them—could not satisfy linear/nonlinear and inequality/equality constraints, which indicates that maybe there is no feasible sample in the initial population. Therefore, constraint satisfaction is the first issue for uniform designs in constrained experimental domains.

Based on this analysis, we establish the first fitness function. Since the degree of constraint violation can directly measure the feasibility of a sample, minimizing the degree of constraint violation is an efficient way to motivate a sample toward the constrained experimental domain. The degree of constraint violation of a sample \vec{x}_i ($i \in \{1, 2, \dots, NP\}$) on the k th constraint is computed by the following equation:

$$\begin{cases} G_k(\vec{x}_i) = \max\{0, g_k(\vec{x}_i)\}, & 1 \leq k \leq p \\ H_k(\vec{x}_i) = \max\{0, |h_k(\vec{x}_i)| - \delta\}, & 1 \leq k \leq q \end{cases} \quad (6)$$

where $g_k(\vec{x}_i)$ denotes the k th inequality constraint, p is the number of inequality constraints, $G_k(\vec{x}_i)$ represents the degree of constraint violation on the k th inequality constraint, $h_k(\vec{x}_i)$ denotes the k th equality constraint, q is the number of equality constraints, δ is the tolerance value to relax the equality

constraint to a certain extent, and $H_k(\vec{x}_i)$ represents the degree of constraint violation on the k th equality constraint.

Then, the degree of constraint violation of \vec{x}_i on all constraints can be expressed as follows:

$$FF_1(\vec{x}_i) = \sum_{k=1}^p G_k(\vec{x}_i) + \sum_{k=1}^q H_k(\vec{x}_i). \quad (7)$$

In this paper, (7) is considered to be the first fitness function for the purpose of constraint satisfaction. Note that the smaller the value of FF_1 , the better the performance of a sample. Moreover, $FF_1(\vec{x}_i) = 0$ means that \vec{x}_i is a feasible sample. By minimizing (7), the degree of constraint violation of the population will decrease gradually, and more and more samples in the population will become feasible. As a result, a predefined number of samples can be obtained in the constrained experimental domain.

The ultimate purpose of uniform designs in constrained experimental domains is to uniformly spread the samples throughout the constrained experimental domain. Therefore, the second issue is related to the uniform distribution of the obtained samples in the constrained experimental domain.

Chen *et al.* [36] proposed a heuristic method, called discrete particle swarm optimization, for constructing uniform designs in constrained experimental domains. In this method, each individual in the population is an $NP \times D$ array and the central composite discrepancy (CCD) [49] is used to measure the uniformity of an individual. However, it is an NP-hard problem to search for the optimal design based on the CCD criterion. Although a discrete version of CCD can save the computational cost, it is still very time-consuming. For example, the uniform design of the new type of automotive crash box introduced in Section II includes 14 decision variables. If a population consists of 50 samples, then the method in [36] will take about 3×10^8 years to produce the optimal uniform design in the constrained experimental domain. Obviously, it is very hard for this method to be directly applied in practical engineering.

To achieve uniform designs in constrained experimental domains more efficiently, we design a second fitness function, which maximizes the minimum Euclidean distance among the samples in the population. This fitness function is able to make an effective tradeoff between uniform distribution and computational cost, which is implemented as follows.

For each sample $\vec{x}_i (i \in \{1, 2, \dots, NP\})$ in the population, we compute the normalized Euclidean distance from \vec{x}_i to the other samples, and the minimum normalized Euclidean distance is denoted as min_dis_i

$$min_dis_i = \min_{j=1,2,\dots, NP, j \neq i} distance(normalized(\vec{x}_i), normalized(\vec{x}_j)) \quad (8)$$

where $distance(\cdot, \cdot)$ represents the Euclidean distance in the decision space, and $normalized(\vec{x}_i)$ and $normalized(\vec{x}_j)$ mean that each dimension of \vec{x}_i and \vec{x}_j is normalized as follows:

$$normalized(x_{i,k}) = (x_{i,k} - x_k^{lower}) / (x_k^{upper} - x_k^{lower}), \quad k = 1, 2, \dots, D \quad (9)$$

$$normalized(x_{j,k}) = (x_{j,k} - x_k^{lower}) / (x_k^{upper} - x_k^{lower}), \quad k = 1, 2, \dots, D. \quad (10)$$

Afterward, the second fitness function can be expressed as follows:

$$FF_2 = \min_{i=1,2,\dots, NP} min_dis_i. \quad (11)$$

The larger the value of FF_2 , the better the overall uniformity of the population. A smaller value of FF_2 means that some samples in the population are stuck at a subregion of the constrained experimental domain. On the contrary, a larger value of FF_2 means that the samples are distributed relatively uniformly in the constrained experimental domain. The computational time complexity of FF_2 is $O(NP^2)$.

According to the characteristics of uniform designs in constrained experimental domains and the two fitness functions, we divide the whole evolutionary process into two phases. These two phases are elaborated on next.

B. Phase 1

The first phase aims at obtaining a predefined number of feasible samples by minimizing the first fitness function. Note that the first phase should also provide feasible candidate solutions with good diversity for the next phase to achieve the ultimate purpose, i.e., uniform design. In order to keep the diversity of the population, a possible way is to guide the samples toward the constrained experimental domain from different directions. Inspired by Wang and Cai [40], a clustering DE is utilized.

First, population \mathbf{P} including NP samples is clustered into $\lfloor NP/NS \rfloor$ subpopulations, where NS denotes the number of samples in each subpopulation and $\lfloor \cdot \rfloor$ denotes rounding down to the nearest integer. Afterward, DE is employed to evolve each subpopulation. The clustering DE is implemented in the following iterative way.

- Step 1: Randomly generate a reference point \vec{r} from the decision space and normalize \vec{r} according to (9).
- Step 2: Set $i = 1$ and $\mathbf{TP} = \emptyset$. // \mathbf{TP} denotes a temporary population.
- Step 3: Compute the normalized Euclidean distance from \vec{r} to the samples in \mathbf{P} and determine the nearest sample in \mathbf{P} to \vec{r} (denoted as \vec{z}).
- Step 4: Find $(NS - 1)$ samples in \mathbf{P} , which are nearest to \vec{z} . These $(NS - 1)$ samples and \vec{z} form a subpopulation, denoted as \mathbf{SP}_i .
- Step 5: Delete these NS samples from \mathbf{P} .
- Step 6: Generate an offspring subpopulation (denoted as \mathbf{OP}_i) for \mathbf{SP}_i by the mutation and crossover of DE, and implement the selection of DE on \mathbf{SP}_i and \mathbf{OP}_i to update \mathbf{SP}_i . Note that the comparison is based on the first fitness function FF_1 .
- Step 7: Incorporate the updated \mathbf{SP}_i into the temporary population \mathbf{TP} , i.e., $\mathbf{TP} = \mathbf{TP} \cup \mathbf{SP}_i$.
- Step 8: $i = i + 1$. If $i \leq \lfloor NP/NS \rfloor$, then go to step 3; otherwise $\mathbf{P} = \mathbf{P} \cup \mathbf{TP}$.
- Step 9: If each subpopulation has at least $\lfloor NP'/(NP/NS) \rfloor$ feasible samples, then randomly choose $\lfloor NP'/(NP/NS) \rfloor$ feasible samples from each subpopulation to form a new population \mathbf{P}' for the second phase; otherwise go to step 1.

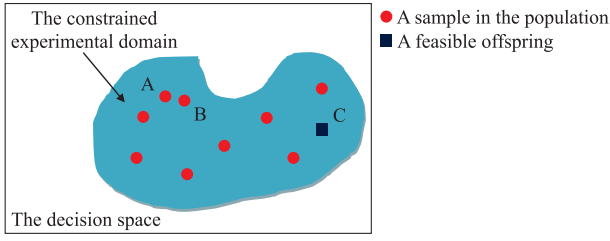


Fig. 4. Example of the replacement strategy. C is a feasible offspring which is added into the population, and A and B are two samples in the population. Since A and B have the minimum normalized Euclidean distance and the second minimum normalized Euclidean distance of A is less than that of B, A is deleted from the population. If the updated population has a better value of FF_2 , then the replacement is successful. Otherwise, the population returns to its original state.

Since different subpopulations converge toward the constrained experimental domain from different directions, in step 9 we choose $\lfloor NP'/(NP/NS) \rfloor$ feasible samples from each subpopulation to guarantee the diversity of P' , where NP' is the population size of P' . In general, $NP' \ll NP$.

C. Phase 2

After the first phase, we obtain the population P' , which contains a predefined number of feasible samples (i.e., NP'). In the second phase, the mutation and crossover of DE are executed to generate NP' offspring for P' . It is noteworthy that maybe some offspring are not in the constrained experimental domain. Therefore, it is necessary to check whether a newly generated offspring is feasible or not by the first fitness function FF_1 . If all the newly generated offspring are infeasible, DE will be implemented continuously until at least one offspring is feasible.

In order to make P' distribute uniformly in the constrained experimental domain, a novel replacement strategy is proposed in this paper. First, the feasible offspring are added to P' one by one. After adding one feasible offspring to P' , we will choose the sample which has the minimum normalized Euclidean distance to the other samples according to (8). If several samples have the same minimum normalized Euclidean distance, then we will consider the second minimum normalized Euclidean distance and so forth. At last, we obtain one individual. Subsequently, this individual is deleted from P' ; thus, the population size of P' is unchanged. If the updated P' has a better value of FF_2 , then the replacement is successful; otherwise, P' returns to its original state. If the successful replacement cannot occur over continuous $count$ times, then the second phase halts. Fig. 4 gives a simple example. Through this replacement strategy, the overall uniformity of P' can be strengthened constantly during the evolution.

The details of the second phase are given as follows.

Step 1: Compute the fitness function FF_2 for P' .

Step 2: Set $k = 0$.

Step 3: Generate NP' offspring for P' by implementing the mutation and crossover of DE.

Step 4: Find the feasible offspring and suppose that there are NS' feasible offspring.

Step 5: If $NS' = 0$, which suggests that there are no feasible offspring, then go to step 3.

Step 6: For $i = 1 : NS'$.

Step 6.1: $Q' = P'$; // Q' denotes the original state of P' .

Step 6.2: Put the i th feasible offspring into P' .

Step 6.3: Choose the sample having the minimum normalized Euclidean distance according to (8) (if several samples have the same minimum normalized Euclidean distance, then select the sample with the second minimum normalized Euclidean distance and so forth) and delete it from P' .

Step 6.4: Compute the fitness function FF_2 for the updated P' .

Step 6.5: If the updated P' has a better value of FF_2 , then it is a successful replacement and $k = 0$; otherwise $P' = Q'$ and $k = k + 1$.

Step 6.6: If $k > count$, break and output P' .

Step 7: Go to step 3.

Remark 1: Based on our introduction, it can be concluded that the above two phases combined with the two fitness functions can not only guide the population toward the constrained experimental domain from different directions quickly, but also have the potential to uniformly scatter the population in the constrained experimental domain.

D. Performance Criterion

With the termination of the second phase, the final population P' can be obtained. In order to evaluate the performance of P' , the maximum distance-based criterion (called MD) is adopted [33]. According to [33], first we need to produce a test set TS , which contains m individuals randomly generated from the constrained experimental domain and m is a very large integer. In this paper, TS is produced by repeatedly implementing the first phase of ToPDE introduced in Section IV-B until m individuals have been obtained in the constrained experimental domain. Afterward, the following procedure is executed.

Step 1: Normalize each sample in P' and each individual in TS according to (9). After the normalization, let $P' = \{\vec{x}'_1, \vec{x}'_2, \dots, \vec{x}'_{NP'}\}$ and $TS = \{\vec{y}'_1, \vec{y}'_2, \dots, \vec{y}'_m\}$.

Step 2: For each $\vec{y}'_j (j = 1, 2, \dots, m)$ in TS , compute the minimum normalized Euclidean distance from \vec{y}'_j to P' : $\min_dis(\vec{y}'_j, P') = \min_{i=1,2,\dots,NP'} distance(\vec{y}'_j, \vec{x}'_i)$.

Step 3: Compute the maximum $\min_dis(\vec{y}'_j, P')$: $MD = \max_{j=1,2,\dots,m} \min_dis(\vec{y}'_j, P')$.

In principle, this performance criterion computes the maximum distance between any $\vec{y}'_j (j = 1, 2, \dots, m)$ in TS and the sample in P' nearest to \vec{y}'_j . A small value of MD implies that the individuals in TS tend to be close to P' , and a large value of MD implies that some subregions are not well covered by P' [33]. Therefore, the smaller the value of MD , the better the performance of a method.

TABLE III
DETAILS OF THE 2-D EXAMPLE

Relationship of the variables	Type of the constraints
$-20 \leq x_1 \leq 20, -10 \leq x_2 \leq 10$	Boundary constraints
$-x_1 + x_2 - 5 \leq 0, x_1^2 + 5x_2^2 - 100 \leq 0,$ $x_1x_2 - 10 \leq 0, x_1x_2 + 4 \leq 0$	Inequality constraints

E. Framework of ToPDE

By integrating the important components introduced in the above four sections, ToPDE for uniform designs in constrained experimental domains works as follows.

- Step 1: Randomly generate an initial population \mathbf{P} from the decision space, which contains NP samples.
 - Step 2: Implement the first phase and obtain another population \mathbf{P}' which contains NP' diversified samples in the constrained experimental domain.
 - Step 3: Implement the second phase to make the NP' samples in \mathbf{P}' distribute uniformly in the constrained experimental domain.
 - Step 4: Evaluate \mathbf{P}' via the performance criterion.
- ToPDE includes six main parameters.
- 1) NP : The size of population \mathbf{P} in the first phase.
 - 2) NS : The size of each subpopulation in the first phase.
 - 3) NP' : The size of population \mathbf{P}' in the second phases.
 - 4) $count$: The maximum tolerance value for the unsuccessful replacement in the second phase.
 - 5) F : The scaling factor of DE.
 - 6) CR : The crossover control parameter of DE.

V. EXPERIMENTAL STUDY

In this section, at first a 2-D example is utilized to show how ToPDE works. Afterward, ToPDE is applied to the uniform design of the new type of automotive crash box. In order to further test the performance of ToPDE, it is also applied to five benchmark test problems. Moreover, the performance of ToPDE is compared with that of another state-of-the-art method, called NTMD [33].

A. 2-D Example

A 2-D example is designed to explain the working principle of ToPDE. This example contains two decision variables and, therefore, is easy to visualize. In addition, there are one linear inequality constraint and three nonlinear inequality constraints. The details of this example are given in Table III. Assume that the aim is to attain 20 uniform samples in the constrained experimental domain.

The parameter settings of ToPDE are given in Table IV and they were used in the experiments. For the performance criterion, the size of the test set was set to 10 000. After the initialization, the first phase of ToPDE was executed. The aim of this phase is to guide the population toward the constrained experimental domain from different directions promptly. Following the steps of the first phase introduced in Section IV-B, a reference point was randomly selected, and five (i.e., $\lfloor NP/NS \rfloor$) subpopulations were generated by

TABLE IV
PARAMETER SETTINGS OF TOPDE FOR THE 2-D EXAMPLE

Parameter	Value	Parameter	Value
NP	100	$count$	500
NS	20	F	0.9
NP'	20	CR	0.9

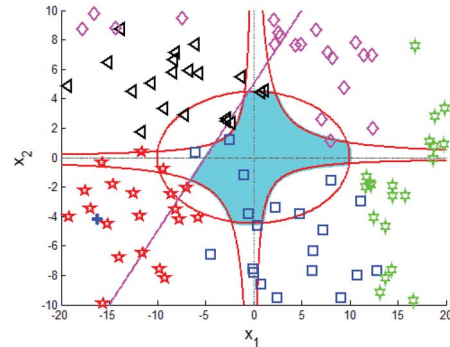


Fig. 5. Initial distribution of the five subpopulations. The blue cross is the reference point, the five subpopulations are depicted with five different colors and shapes, and the cyan part is the constrained experimental domain.

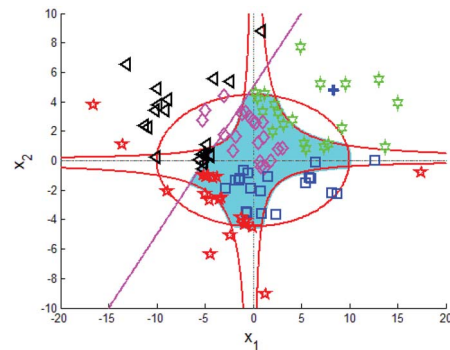


Fig. 6. End of the first phase. Under this condition, each subpopulation has at least four feasible samples.

the clustering DE. The initial distribution of the five subpopulations is shown in Fig. 5. From Fig. 5, we can see that the five subpopulations, depicted with five different colors and shapes, are located at different subregions and represent different search directions. If each subpopulation has at least four [i.e., $\lfloor NP/(NP/NS) \rfloor$] feasible samples, then the first phase terminates. The end of the first phase is shown in Fig. 6. Afterward, four feasible samples were randomly chosen from each subpopulation, and these 20 feasible samples formed the initial population of the second phase, which is shown in Fig. 7.

Subsequently, the second phase was triggered, with the purpose of motivating the 20 feasible samples to uniformly cover the constrained experimental domain. The final population with the end of the second phase is shown in Fig. 8. From Fig. 8, it is clear that ToPDE has the capability to uniformly distribute these 20 feasible samples in the constrained experimental domain.

Moreover, we independently ran ToPDE 50 times for this 2-D example. The MD values over 50 runs are summarized in Table V. From Table V, it can be seen that ToPDE is able to

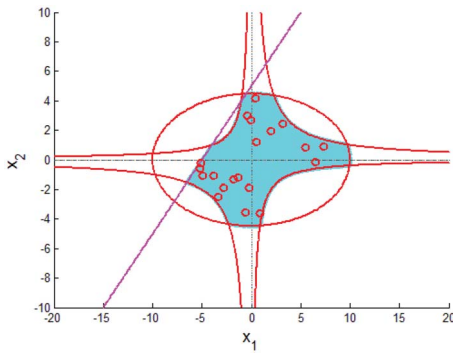


Fig. 7. Initial population of the second phase, which contains 20 feasible samples chosen from the five subpopulations.

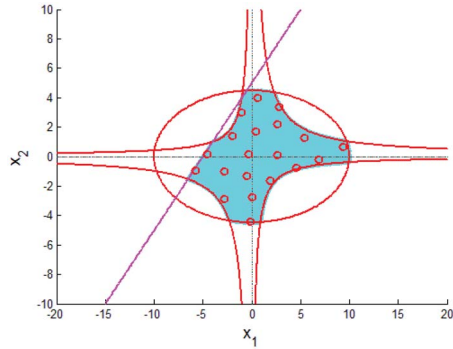


Fig. 8. Final population of the second phase, in which the 20 feasible samples are uniformly distributed in the constrained experimental domain.

TABLE V
EXPERIMENTAL RESULTS OF TOPDE OVER
50 RUNS FOR THE 2-D EXAMPLE

	MD	MD	MD	MD	MD
1	0.0634	11	0.0647	21	0.0584
2	0.0634	12	0.0625	22	0.0585
3	0.0594	13	0.0617	23	0.0609
4	0.0637	14	0.0621	24	0.0651
5	0.0627	15	0.0645	25	0.0603
6	0.0643	16	0.0681	26	0.0581
7	0.0693	17	0.1014	27	0.0772
8	0.0708	18	0.0612	28	0.0577
9	0.0657	19	0.0599	29	0.0646
10	0.0621	20	0.0639	30	0.0636
				40	0.0607
				50	0.0596
Mean Value±Std Dev: 0.0635±0.0066					

consistently provide very small *MD* values, which verifies the stable performance of TopDE.

B. Uniform Design of the New Type of Automotive Crash Box

Section II has introduced a 14-dimensional uniform design of a new type of automotive crash box, the detailed description of which is given in Tables I and II. Assume that the aim is to obtain 100 uniform samples in the constrained experimental domain which is subject to one linear equality constraint and 17 nonlinear inequality constraints.

TopDE was applied to solve this problem. In order to verify its effectiveness, the performance of TopDE was compared with that of NTMD [33]. Previously, NTMD was proposed for the uniform design of a highly constrained mixture experiment which has one linear equality constraint and several inequality

TABLE VI
PARAMETER SETTINGS OF TOPDE FOR THE UNIFORM DESIGN
OF THE NEW TYPE OF AUTOMOTIVE CRASH BOX

Parameter	Value	Parameter	Value
<i>NP</i>	200	<i>count</i>	500
<i>NS</i>	20	<i>F</i>	0.9
<i>NP'</i>	100	<i>CR</i>	0.9

TABLE VII
EXPERIMENTAL RESULTS OF TOPDE OVER 50 RUNS FOR THE UNIFORM
DESIGN OF THE NEW TYPE OF AUTOMOTIVE CRASH BOX

	MD	MD	MD	MD	MD
1	0.6188	11	0.5706	21	0.5490
2	0.5732	12	0.6061	22	0.5919
3	0.6117	13	0.5809	23	0.5818
4	0.5880	14	0.5343	24	0.6042
5	0.5678	15	0.5689	25	0.5352
6	0.5865	16	0.5954	26	0.6042
7	0.5525	17	0.6276	27	0.5339
8	0.6240	18	0.5678	28	0.6165
9	0.6135	19	0.5724	29	0.5585
10	0.5798	20	0.5731	30	0.6051
				40	0.6038
				50	0.5711
Mean Value±Std Dev: 0.5856±0.0290					

TABLE VIII
EXPERIMENTAL RESULTS OF NTMD OVER 50 RUNS FOR THE UNIFORM
DESIGN OF THE NEW TYPE OF AUTOMOTIVE CRASH BOX

	MD	MD	MD	MD	MD
1	0.7142	11	0.7472	21	0.7328
2	0.7734	12	0.7673	22	0.7251
3	0.6612	13	0.7267	23	0.6721
4	0.7136	14	0.7313	24	0.7355
5	0.6913	15	0.7610	25	0.7656
6	0.6745	16	0.6935	26	0.7265
7	0.7144	17	0.7547	27	0.7645
8	0.7237	18	0.6924	28	0.6523
9	0.6667	19	0.7063	29	0.7481
10	0.7410	20	0.7597	30	0.6795
				40	0.7463
				50	0.7382
Mean Value±Std Dev: 0.7149±0.0339					

constraints. It is worth noting that all the decision variables are involved in the linear equality constraint. The implementation of NTMD is given in the Appendix, in which step 4 is used to deal with the linear equality constraint and step 5 is employed to handle other inequality constraints.

Different from [33], some of the decision variables in the uniform design of the new type of automotive crash box are not involved in the linear equality constraint. As shown in Table I, five decision variables (i.e., $x_{10}, x_{11}, \dots, x_{14}$) are not included in the linear equality constraint. Thus, NTMD cannot be directly applied to the uniform design of the new type of automotive crash box. Next, we make a simple revision of step 4 of NTMD.

Step 4: Produce a number of samples, in which x_1, x_2, \dots, x_9 satisfy the linear equality constraint by implementing (14)–(19), and $x_{10}, x_{11}, \dots, x_{14}$ are uniformly sampled by taking advantage of the Latin square design.

In the experiments, 50 independent runs were implemented for TopDE and NTMD. The parameter settings of TopDE are summarized in Table VI. For the performance criterion, the size of the test set was set to 50 000. The experimental results of TopDE and NTMD are listed in Tables VII and VIII, respectively.

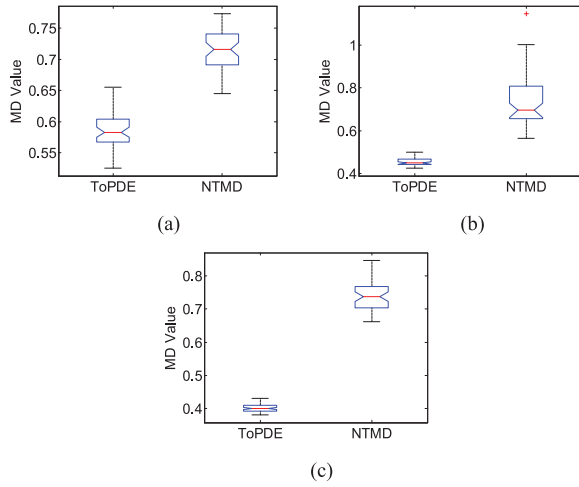


Fig. 9. Box plots of the experimental results provided by ToPDE and NTMD for the uniform designs of the new type of automotive crash box, G04, and G09. (a) New type of automotive crash box. (b) G04. (c) G09.

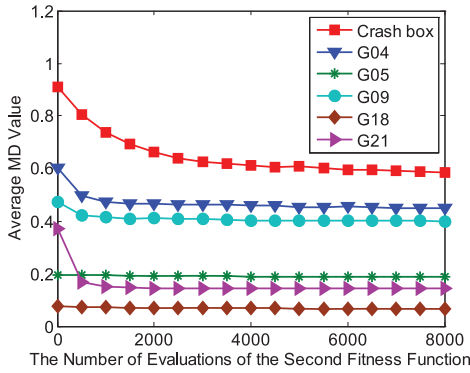


Fig. 10. Evolution of the mean MD values derived from ToPDE over 50 runs versus the number of evaluations of the second fitness function (i.e., FF_2) on the new type of automotive crash box, G04, G05, G09, G18, and G21.

From Tables VII and VIII, we can see that the mean MD values of ToPDE and NTMD are 0.5856 and 0.7149, respectively. Therefore, the performance of ToPDE is on average 18.09% better than that of NTMD. Furthermore, Fig. 9(a) provides the box plot of the experimental results of ToPDE and NTMD. The convergence curve of the average MD value derived from ToPDE over 50 runs is presented in Fig. 10. In Fig. 10, the iteration will terminate if the number of evaluations of the second fitness function (i.e., FF_2) has reached 8000. As shown in Fig. 10, ToPDE converges after about 3000 evaluations of the second fitness function.

The above experimental results reveal that ToPDE is more effective than NTMD for the uniform design of the new type of automotive crash box.

C. Five Benchmark Test Problems

In order to further verify the effectiveness of ToPDE on other uniform designs in constrained experimental domains, five benchmark test problems (called G04, G05, G09, G18, and G21) are selected from “the special session and competition on evolutionary constrained real parameter single-objective optimization” of the 2006 IEEE Congress on Evolutionary Computation [50]. Note that the purpose of this paper is to

study how to uniformly sample over the constrained experimental domain. Consequently, we only exploit the decision variables and constraints of these test problems, and their objective functions are ignored. The decision variables and constraints of G04, G05, G09, G18, and G21 are summarized in Table IX. In Table IX, D is the number of decision variables, LI is the number of linear inequality constraints, NI is the number of nonlinear inequality constraints, LE is the number of linear equality constraints, NE is the number of nonlinear equality constraints, and θ is the estimated feasibility ratio between the constrained experimental domain and the decision space. The characteristics of these five test problems can be briefly summarized as follows.

- 1) G04, G09, and G18 only contain inequality constraints, and G05 and G21 include both equality and inequality constraints.
- 2) G04 has a moderate feasibility ratio (i.e., 51.1230%) and G09 has a small feasibility ratio (i.e., 0.5121%). In particular, G05, G18, and G21 are highly constrained problems, the feasibility ratios of which are extremely small and approximate to zero.¹

With respect to these five test problems, 50 independent runs were implemented for ToPDE and NTMD. The parameter settings of ToPDE were the same as in Section V-B. For the performance criterion, the size of the test set was set to 50 000. Tables X and XI summarize the experimental results of ToPDE and NTMD on G04, and Tables XII and XIII summarize the experimental results of ToPDE and NTMD on G09. In addition, the box plots for the performance comparison between ToPDE and NTMD on G04 and G09 are given in Fig. 9(b) and (c), respectively.

Regarding G04, the mean MD values provided by ToPDE and NTMD are 0.4543 and 0.7310, respectively, which means that ToPDE is on average 37.85% better than NTMD. Fig. 9(b) describes the distribution of the MD values over 50 independent runs. It is obvious from Fig. 9(b) that the worst MD value derived from ToPDE is even better than the best MD value provided by NTMD.

For G09, the mean MD values of ToPDE and NTMD are 0.4012 and 0.7382, respectively, which signifies that ToPDE achieves 45.65% performance improvement. It can be observed again from Fig. 9(c) that the worst MD value of ToPDE is even better than the best MD value of NTMD.

In terms of G05, G18, and G21, the average and standard deviation of the MD values obtained by ToPDE and NTMD over 50 runs are presented in Table XIV. According

¹For a real-world industrial application, maybe an algorithm cannot find any feasible sample. There are two reasons: 1) there does not exist any region satisfying all constraints and 2) the constrained experimental domain occupies a very small proportion of the decision space, such as G05, G18, and G21. To the best of our knowledge, there is no ready answer for the first scenario. However, for the second scenario, a commonly used method is to relax an equality constraint to a certain degree as in (6). Note that, usually inequality constraints are not relaxed. In (6), δ is a very small positive number and set to 0.0001 in this paper. This manner slightly enlarges the constrained experimental domain and enables an algorithm to enter the constrained experimental domain more easily. In addition, the power of an optimization algorithm also plays a crucial role in searching for the feasible samples in the second scenario.

TABLE IX
DETAILS OF G04, G05, G09, G18, AND G21

Problem	Decision Variable and Constraint	D	LI	NI	LE	NE	θ
G04	$g_1(\vec{x}) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0$ $g_2(\vec{x}) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0$ $g_3(\vec{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0$ $g_4(\vec{x}) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0$ $g_5(\vec{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0$ $g_6(\vec{x}) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0$ $78 \leq x_1 \leq 102, 33 \leq x_2 \leq 45, \text{ and } 27 \leq x_3, x_4, x_5 \leq 45$	5	0	6	0	0	51.1230%
G05	$g_1(\vec{x}) = -x_4 + x_3 - 0.55 \leq 0, g_2(\vec{x}) = -x_3 + x_4 - 0.55 \leq 0$ $h_1(\vec{x}) = 1000 \sin(-x_3 - 0.25) + 1000 \sin(-x_4 - 0.25) + 894.8 - x_1 = 0$ $h_2(\vec{x}) = 1000 \sin(x_3 - 0.25) + 1000 \sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0$ $h_3(\vec{x}) = 1000 \sin(x_4 - 0.25) + 1000 \sin(x_4 - x_3 - 0.25) + 1294.8 = 0$ $0 \leq x_1 \leq 1200, 0 \leq x_2 \leq 1200, -0.55 \leq x_3 \leq 0.55, \text{ and } -0.55 \leq x_4 \leq 0.55$	4	2	0	0	3	0.0000%
G09	$g_1(\vec{x}) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0$ $g_2(\vec{x}) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0$ $g_3(\vec{x}) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0$ $g_4(\vec{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0$ $-10 \leq x_1, x_2, \dots, x_7 \leq 10$	7	0	4	0	0	0.5121%
G18	$g_1(\vec{x}) = x_3^2 + x_4^2 - 1 \leq 0, g_2(\vec{x}) = x_9^2 - 1 \leq 0$ $g_3(\vec{x}) = x_5^2 + x_6^2 - 1 \leq 0, g_4(\vec{x}) = x_1^2 + (x_2 - x_9)^2 - 1 \leq 0$ $g_5(\vec{x}) = (x_1 - x_5)^2 + (x_2 - x_6)^2 - 1 \leq 0, g_6(\vec{x}) = (x_1 - x_7)^2 + (x_2 - x_8)^2 - 1 \leq 0$ $g_7(\vec{x}) = (x_3 - x_5)^2 + (x_4 - x_6)^2 - 1 \leq 0, g_8(\vec{x}) = (x_3 - x_7)^2 + (x_4 - x_8)^2 - 1 \leq 0$ $g_9(\vec{x}) = x_7^2 + (x_8 - x_9)^2 - 1 \leq 0, g_{10}(\vec{x}) = x_2x_3 - x_1x_4 \leq 0$ $g_{11}(\vec{x}) = -x_3x_9 \leq 0, g_{12}(\vec{x}) = x_3x_9 \leq 0, g_{13}(\vec{x}) = x_6x_7 - x_5x_8 \leq 0$ $-10 \leq x_1, x_2, \dots, x_8 \leq 10 \text{ and } 0 \leq x_9 \leq 20$	9	0	13	0	0	0.0000%
G21	$g_1(\vec{x}) = -x_1 + 35x_2^{0.6} + 35x_3^{0.6} \leq 0$ $h_1(\vec{x}) = -300x_3 + 7500x_5 - 7500x_6 - 25x_4x_5 + 25x_4x_6 + x_3x_4 = 0$ $h_2(\vec{x}) = 100x_2 + 155.365x_4 + 2500x_7 - x_2x_4 - 25x_4x_7 - 15536.5 = 0$ $h_3(\vec{x}) = -x_5 + \ln(-x_4 + 900) = 0, h_4(\vec{x}) = -x_6 + \ln(x_4 + 300) = 0$ $h_5(\vec{x}) = -x_7 + \ln(-2x_4 + 700) = 0$ $0 \leq x_1 \leq 1000, 0 \leq x_2, x_3 \leq 40, 100 \leq x_4 \leq 300, 6.3 \leq x_5 \leq 6.7, 5.9 \leq x_6 \leq 6.4, \text{ and } 4.5 \leq x_7 \leq 6.25$	7	0	1	0	5	0.0000%

to our observation, NTMD cannot find any feasible solution even after running one week. Therefore, in Table XIV, “NA” denotes the experimental results of NTMD are not available. The reason why NTMD cannot find any feasible solution on G05, G18, and G21 can be explained as follows. NTMD is very suitable for dealing with one linear equality constraint. However, G05, G18, and G21 do not involve any linear equality constraint. Under this condition, NTMD is equivalent to a traditional uniform design method as pointed out in Section I. Note that when handling other constraints, a traditional uniform design method generates a large number of samples and retains only those samples satisfying all constraints. Therefore, it is essentially a brute-force method. Due to the fact that the feasibility ratios of G05, G18, and G21 are extremely small, it is not difficult to understand that NTMD fails to find any feasible solution. As shown in Table XIV, TopPDE achieves quite good performance since the mean MD values are consistently less than 0.2.

TABLE X
EXPERIMENTAL RESULTS OF TOPPDE OVER 50 RUNS ON G04

	MD	MD	MD	MD	MD	MD	MD	MD	MD
1	0.4813	11	0.4762	21	0.4703	31	0.4405	41	0.4789
2	0.4455	12	0.4750	22	0.4375	32	0.4866	42	0.4281
3	0.4310	13	0.4427	23	0.4722	33	0.4465	43	0.4565
4	0.4454	14	0.4886	24	0.4575	34	0.4449	44	0.4466
5	0.4406	15	0.4275	25	0.4644	35	0.4414	45	0.4605
6	0.4369	16	0.4647	26	0.4782	36	0.4353	46	0.4606
7	0.4643	17	0.4681	27	0.4265	37	0.4784	47	0.4989
8	0.4331	18	0.4421	28	0.4950	38	0.4459	48	0.4571
9	0.4246	19	0.4533	29	0.4528	39	0.4502	49	0.4432
10	0.4465	20	0.4435	30	0.4435	40	0.4310	50	0.4536
Mean Value±Std Dev: 0.4543±0.0190									

The convergence curves of the average MD value derived from TopPDE over 50 runs on these five test problems are given in Fig. 10. From Fig. 10, it is interesting to see that TopPDE exhibits very fast convergence speed. In particular, for G05

TABLE XI
EXPERIMENTAL RESULTS OF NTMD OVER 50 RUNS ON G04

	<i>MD</i>		<i>MD</i>		<i>MD</i>		<i>MD</i>		<i>MD</i>
1	0.6700	11	0.7938	21	0.6690	31	0.6896	41	0.6230
2	0.7092	12	0.8518	22	0.6377	32	0.5905	42	0.5777
3	0.7112	13	0.6836	23	1.0035	33	0.7344	43	0.6838
4	0.6903	14	0.7976	24	0.5653	34	0.8134	44	0.6965
5	0.6882	15	0.5988	25	0.8874	35	0.8488	45	0.8155
6	0.8097	16	0.7760	26	0.6888	36	0.6566	46	0.8581
7	0.7851	17	0.6961	27	0.6135	37	0.6842	47	0.5957
8	0.7876	18	0.8370	28	0.6958	38	0.5849	48	0.9569
9	0.6526	19	0.7081	29	0.5899	39	0.7531	49	0.6053
10	0.8274	20	1.1475	30	0.6761	40	0.6874	50	0.8443
Mean Value±Std Dev: 0.7310±0.1177									

TABLE XII
EXPERIMENTAL RESULTS OF TOPDE OVER 50 RUNS ON G09

	<i>MD</i>		<i>MD</i>		<i>MD</i>		<i>MD</i>		<i>MD</i>
1	0.4013	11	0.3999	21	0.3924	31	0.4070	41	0.3919
2	0.4094	12	0.4174	22	0.3950	32	0.4072	42	0.3802
3	0.3987	13	0.3921	23	0.3874	33	0.3978	43	0.4165
4	0.4309	14	0.4129	24	0.3892	34	0.3836	44	0.3862
5	0.4060	15	0.3931	25	0.3996	35	0.4179	45	0.4112
6	0.3879	16	0.3982	26	0.3941	36	0.4091	46	0.4099
7	0.3911	17	0.3997	27	0.4015	37	0.3988	47	0.4094
8	0.3810	18	0.4126	28	0.4092	38	0.3925	48	0.3946
9	0.3883	19	0.3951	29	0.4056	39	0.4145	49	0.4167
10	0.4167	20	0.4090	30	0.3838	40	0.3893	50	0.4263
Mean Value±Std Dev: 0.4012±0.0120									

TABLE XIII
EXPERIMENTAL RESULTS OF NTMD OVER 50 RUNS ON G09

	<i>MD</i>		<i>MD</i>		<i>MD</i>		<i>MD</i>		<i>MD</i>
1	0.7430	11	0.8474	21	0.6789	31	0.7315	41	0.6740
2	0.7603	12	0.7721	22	0.7153	32	0.6705	42	0.6751
3	0.7075	13	0.6615	23	0.6965	33	0.7353	43	0.6962
4	0.6724	14	0.7408	24	0.7399	34	0.8025	44	0.7036
5	0.7249	15	0.7770	25	0.7391	35	0.8350	45	0.8246
6	0.7648	16	0.7530	26	0.7420	36	0.6949	46	0.7602
7	0.7291	17	0.7873	27	0.7101	37	0.7721	47	0.7218
8	0.7618	18	0.7574	28	0.7785	38	0.7114	48	0.6862
9	0.8189	19	0.7138	29	0.7247	39	0.7685	49	0.6804
10	0.6844	20	0.7745	30	0.6789	40	0.7221	50	0.7667
Mean Value±Std Dev: 0.7382±0.0460									

TABLE XIV
AVERAGE AND STANDARD DEVIATION OF THE *MD* VALUES (DENOTED AS “MEAN VALUE” AND “STD DEV”) OBTAINED BY TOPDE AND NTMD OVER 50 RUNS ON G05, G18, AND G21

Problem	Method	Mean Value±Std Dev
G05	TopPDE	0.1869±0.0648
	NTMD	<i>NA</i>
G18	TopPDE	0.0679±0.0062
	NTMD	<i>NA</i>
G21	TopPDE	0.1223±0.0880
	NTMD	<i>NA</i>

and G18, the performance of TopPDE is nearly stable at the beginning of the second phase.

The above experimental results demonstrate that TopPDE is also an effective method for other uniform designs in constrained experimental domains, and that TopPDE is not dependent on the type and number of constraints. In the case of moderate and small constrained experimental domains (such as G04 and G09), the performance of TopPDE is significantly better than that of NTMD. Moreover, it seems that NTMD is not capable of coping with the uniform designs in the highly

constrained experimental domains without any linear equality constraint (such as G05, G18, and G21).

Remark 2: As pointed out by Borkowski and Piepel [33], for the high-dimensional constrained experimental domain, distance-based methods such as [11], [51], and [52] tend to move samples on or near the boundary, and the interior region is thus not effectively covered. Our proposed TopPDE is also a distance-based method. However, it still works well on the high-dimensional constrained experimental domain. The above phenomenon can be illustrated as follows. First, the methods in [11], [51], and [52] maximize the minimum distance between two samples. It is a local manner to improve the uniformity. Note that although the performance of these two samples can be improved, the overall uniformity may deteriorate. In contrast, TopPDE maximizes the minimum distance of the entire population in a global manner, which pushes the entire population as far apart as possible. Second, the methods in [11], [51], and [52] produce new samples by local search algorithms. As a result, their capability to explore the previously uncovered areas is limited in the high-dimensional constrained experimental domain. On the contrary, TopPDE designs a population-based global search algorithm, which is much more powerful, to produce samples in some promising areas of the high-dimensional constrained experimental domain.

VI. DISCUSSION

In this section, we investigate the advantage of TopPDE over NTMD, the scalability to the sample size, the effect of the disconnected constrained experimental domain, and the effectiveness of the mutation operator of TopPDE.

A. Advantage of TopPDE Over NTMD

The experimental results in Sections V-B and V-C have corroborated the superior performance of TopPDE. In this section, the advantage of TopPDE over NTMD is further analyzed by taking G04 as an example. Figs. 11 and 12 show the distributions in the $x_1 - x_4$ space and in the $x_3 - x_4$ space produced by the test set, the final population of TopPDE in a typical run, and the final population of NTMD in a typical run, respectively. In this paper, a typical run means the run in which the *MD* value ranks 26th (from worst to best) among 50 runs.

As mentioned previously, the general purpose of uniformly sampling over the constrained experimental domain is to provide adequate support for developing the surrogate model. Therefore, it is very important to include the samples on the boundary as well as in the interior of the constrained experimental domain [13], [33]. As shown in Fig. 11, NTMD pays less attention to the boundary of the constrained experimental domain and is very likely to miss some parts of the constrained experimental domain in the $x_1 - x_4$ space. In contrast, TopPDE has the advantage of being able to not only place a number of samples exactly on the boundary of the constrained experimental domain, but also cover the interior of the constrained experimental domain with a lot of samples.

Fig. 12 suggests that TopPDE still maintains good space-filling performance in the $x_3 - x_4$ space. However, the samples

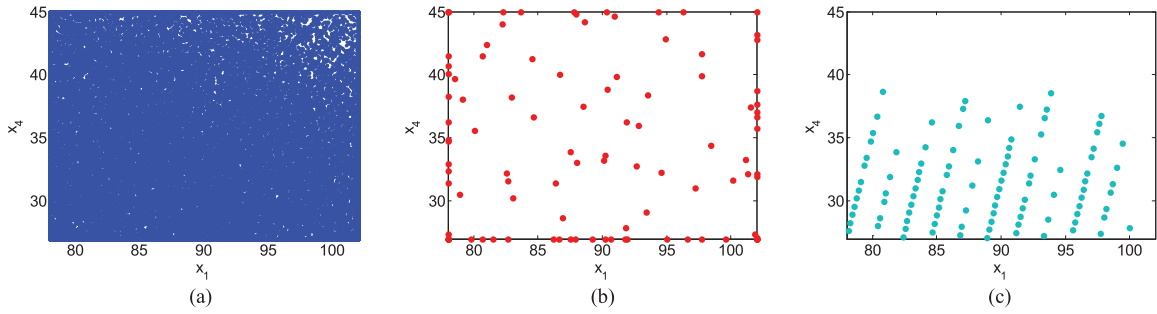


Fig. 11. Distributions in the $x_1 - x_4$ space on G04. Distribution of the (a) test set containing 50 000 points, (b) final population of ToPDE in a typical run, and (c) final population of NTMD in a typical run.

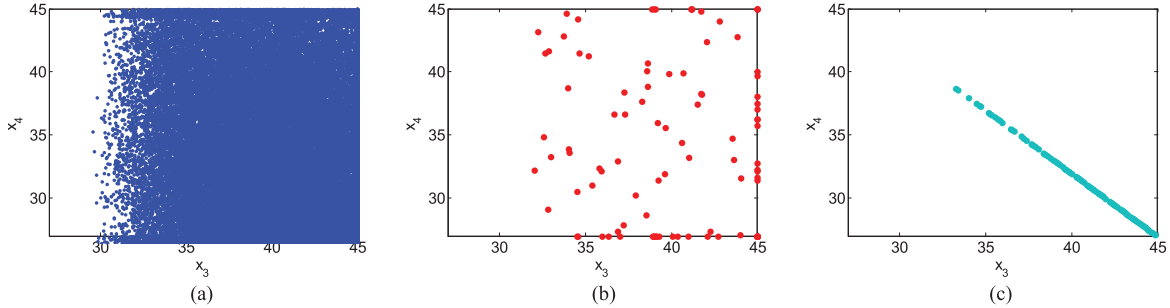


Fig. 12. Distributions in the $x_3 - x_4$ space on G04. Distribution of the (a) test set containing 50 000 points, (b) final population of ToPDE in a typical run, and (c) final population of NTMD in a typical run.

TABLE XV

EXPERIMENTAL RESULTS OF TOPDE AND NTMD OVER 50 RUNS FOR THE NEW TYPE OF AUTOMOTIVE CRASH BOX, G04, AND G09 WITH VARYING SAMPLE SIZES. “MEAN VALUE” AND “STD DEV” INDICATE THE AVERAGE AND STANDARD DEVIATION OF THE MD VALUE, RESPECTIVELY. PERCENTAGES IN PARENTHESES DENOTE THE IMPROVEMENT RATES

Problem	Method	Sample Size			
		50 Mean Value±Std Dev (Improvement Rate)	100 Mean Value±Std Dev (Improvement Rate)	200 Mean Value±Std Dev (Improvement Rate)	300 Mean Value±Std Dev (Improvement Rate)
Crash box	ToPDE	0.6890±0.0447	0.5856±0.0290	0.5144±0.0301	0.4793±0.0279
	NTMD	0.8002±0.0477 (13.90%)	0.7149±0.0339 (18.09%)	0.6457±0.0297 (20.33%)	0.6067±0.0273 (21.00%)
G04	ToPDE	0.5520±0.0237	0.4543±0.0190	0.3782±0.0106	0.3449±0.0106
	NTMD	0.7380±0.0753 (25.20%)	0.7310±0.1177 (37.85%)	0.6752±0.1049 (43.99%)	0.6312±0.1148 (45.36%)
G09	ToPDE	0.4604±0.0210	0.4012±0.0120	0.3512±0.0150	0.3260±0.0094
	NTMD	1.5780±0.1270 (70.82%)	0.7382±0.0460 (45.65%)	0.6665±0.0621 (47.31%)	0.6201±0.0527 (47.43%)

resulting from NTMD are nearly located on a line segment. It is because the nonlinear interaction between x_3 and x_4 cannot be identified by NTMD.

B. Scalability to the Sample Size

Prior to establishing the surrogate model, a lot of computational resources and costs must be spent evaluating the performance of the obtained samples by physical examples/simulations. For example, for establishing the surrogate model of a new type of automotive crash box, crashworthiness criteria such as energy absorption and peak force should be evaluated. Note that this process is computationally expensive; therefore, a moderate sample size is usually recommended (e.g., 100).

However, from the viewpoint of algorithm design, one may still be interested in the scalability of ToPDE and NTMD to the sample size. To this end, we tested four different sample

sizes: 50, 100, 200, and 300. The average and standard deviation of the MD values resulting from ToPDE and NTMD are given in Table XV and Fig. 13 presents the visualized comparison. Since NTMD fails to find any feasible solution for G05, G18, and G21, only the experimental result of the new type of automotive crash box, G04, and G09 are summarized in Table XV. It is noteworthy that Table XV also reports the improvement rate achieved by ToPDE against NTMD.

The first observation from Table XV is that the mean value of ToPDE and NTMD gradually decreases with the increase of the sample size. In addition, in terms of the improvement rate, ToPDE has the increasing advantage over NTMD with the increase of the sample size on the three problems, except that for G09 ToPDE achieves the highest improvement rate when the sample size is equal to 50.

The above comparison verifies that ToPDE consistently outperforms NTMD regardless of the sample size.

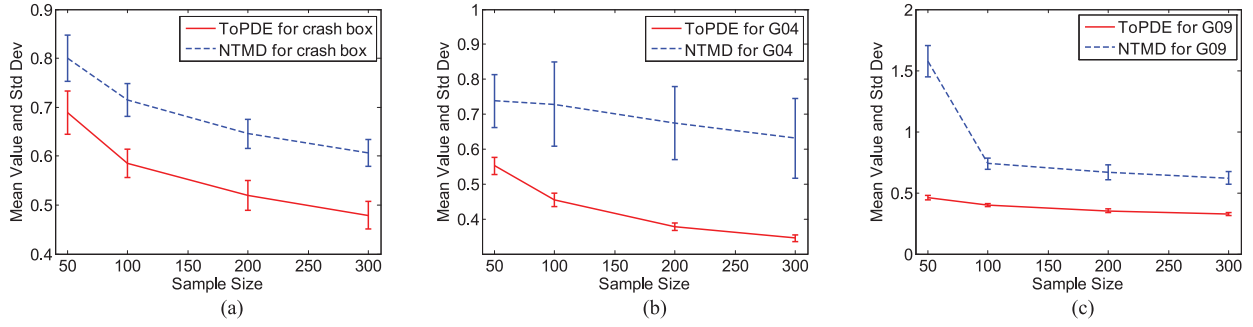


Fig. 13. Average and standard deviation of the MD values derived from ToPDE and NTMD with four different sample sizes, i.e., 50, 100, 200, and 300. (a) New type of automotive crash box. (b) G04. (c) G09.

TABLE XVI

EXPERIMENTAL RESULTS OF TOPDE AND TOPDE_1 OVER 50 RUNS. “MEAN VALUE” AND “STD DEV” INDICATE THE AVERAGE AND STANDARD DEVIATION OF THE MD VALUE, RESPECTIVELY. WILCOXON’S RANK SUM TEST AT A 0.05 SIGNIFICANCE LEVEL IS PERFORMED BETWEEN TOPDE AND TOPDE_1

Problem	Method	Mean Value±Std Dev
Crash box	ToPDE	0.5856±0.0290+
	ToPDE_1	0.6352±0.0813
G04	ToPDE	0.4543±0.0190≈
	ToPDE_1	0.4588±0.0177
G05	ToPDE	0.1869±0.0648+
	ToPDE_1	0.3140±0.1026
G09	ToPDE	0.4012±0.0120≈
	ToPDE_1	0.4004±0.0131
G18	ToPDE	0.0679±0.0062≈
	ToPDE_1	0.0721±0.0066
G21	ToPDE	0.1223±0.0880
	ToPDE_1	NA

“+” and “≈” denote that the performance of ToPDE is better than and similar to that of ToPDE_1, respectively

C. Effect of the Disconnected Constrained Experimental Domain

In the previous experiments, the constrained experimental domain was connected. A question which arises naturally is whether ToPDE is also applicable to a uniform design in a disconnected constrained experimental domain, which is composed of several disjointed areas. To ascertain the effect of the disconnected constrained experimental domain on the performance of ToPDE, we devised the following test problem:

$$\begin{aligned}
 g_1(\vec{x}) &= (x_1 + 4)^2 + (x_2 - 4)^2 - 3 \leq 0 \text{ or} \\
 g_2(\vec{x}) &= (x_1 - 4)^2 + (x_2 + 4)^2 - 3 \leq 0 \text{ or} \\
 g_3(\vec{x}) &= (x_1 + 2)^2 + (x_2 + 2)^2 - 3 \leq 0 \text{ or} \\
 g_4(\vec{x}) &= (x_1 - 2)^2 + (x_2 - 2)^2 - 3 \leq 0 \\
 -8 &\leq x_1, x_2 \leq 8.
 \end{aligned} \tag{12}$$

Based on this test problem, we also designed three scenarios, denoted as S1–S3. S1–S3 contain the first two constraints, the first three constraints, and all the constraints, respectively. Therefore, their constrained experimental domains, respectively, consist of two, three, and four disjointed areas, each of which is a circle. In S1–S3, if a sample lies within one of the disjointed areas, then it is feasible.

When dealing with the above three scenarios, the parameter settings of ToPDE were kept untouched and the purpose is to

obtain 100 uniform samples in each scenario. The experimental results of a typical run provided by ToPDE are given in Fig. 14. As shown in Fig. 14, ToPDE achieves quite promising performance in that a good distribution can be maintained in each disjointed area of each scenario. Moreover, based on our observation, each disjointed area nearly has the same number of samples in each scenario, which implies that ToPDE is also able to maintain a good overall distribution over all the disjointed areas. The excellent performance of ToPDE can be attributed to the fact that the first phase of ToPDE guides the population toward the disconnected constrained experimental domain from different directions and effectively keeps the diversity of the population, and that the second phase of ToPDE facilitates the uniform distribution of samples in each disjointed area.

D. Effectiveness of DE/current-to-rand/1 Mutation Operator

In this paper, DE/current-to-rand/1 is adopted as the mutation operator in DE. To study how the performance of ToPDE is affected by the mutation operator, we implemented a variant of ToPDE called ToPDE_1, in which DE/current-to-rand/1 in (3) is replaced by DE/rand/1 in (2). The experimental results of ToPDE and ToPDE_1 are presented in Table XVI for the new type of automotive crash box, G04, G05, G09, G18, and G21, in which “NA” denotes the experimental results are not available due to the fact that ToPDE_1 cannot enter the second phase on G21.

Table XVI indicates that ToPDE performs statistically better than ToPDE_1 on two problems (i.e., the new type of automotive crash box and G05), and performs similar with ToPDE_1 on three problems (i.e., G04, G09, and G18). However, ToPDE_1 cannot surpass ToPDE even on one problem. More importantly, ToPDE_1 cannot succeed in satisfying the stopping criterion of the first phase on G21. The performance superiority of ToPDE can be explained as follows.

- 1) For DE/rand/1, three individuals are randomly selected from the population for mutation and, consequently, the probability that the offspring is better than the parent is low. Thus, for certain problems (such as G21), based on our observation some subpopulations cannot contain a predefined number of the feasible solutions in the end. In addition, for certain problems (such as the new type of automotive crash box and G05), even though

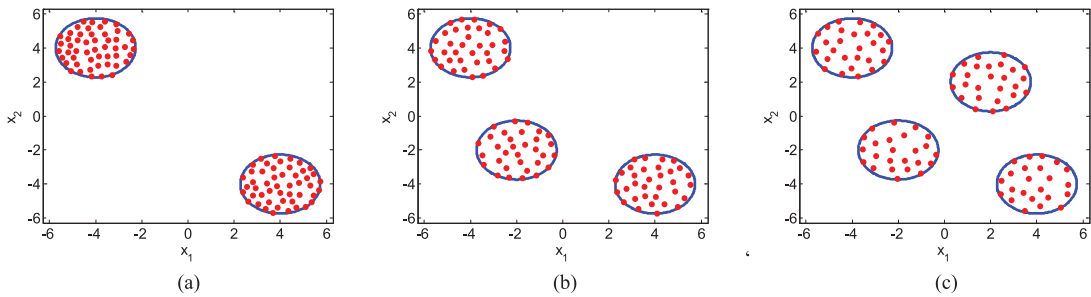


Fig. 14. Experimental results of ToPDE on three scenarios (denoted as S1–S3) with the disconnected constrained experimental domain defined by (12) in a typical run. (a) S1 with two disjoint areas. (b) S2 with three disjoint areas. (c) S3 with four disjoint areas.

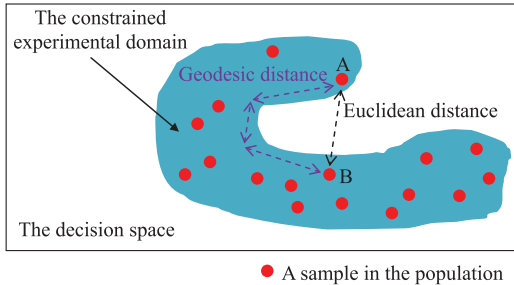


Fig. 15. Example of the geodesic distance and the Euclidean distance between two samples A and B.

the population can enter the second phase, the replacement cannot occur frequently due to the randomness of $DE/rand/1$, which results in the poor performance.

- 2) In contrast, $DE/current-to-rand/1$ generates the offspring around the current individual. As a result, if the population contains some feasible solutions, then ToPDE is very likely to produce the feasible offspring, thus enhancing search efficiency.

The above discussion suggests that $DE/current-to-rand/1$ is a more reasonable choice for uniform designs in constrained experimental domains.

Remark 3: In this paper, ToPDE is a framework for uniform designs in constrained experimental domains and DE is its search engine. The function of DE is to generate offspring. The main reason why we chose DE is twofold.

- 1) The implementation of DE is very simple.
- 2) DE provides various mutation operators and we can choose one of them with the most potential to match the characteristics of an optimization problem.

Up to now, apart from DE, a considerable number of evolutionary algorithm paradigms have been presented, such as genetic algorithm, evolution strategy, evolutionary programming, particle swarm optimization, artificial bee colony algorithm, and so on. In principle, any evolutionary algorithm paradigm can be incorporated into our framework by replacing DE. It is noteworthy that some evolutionary algorithm paradigms (e.g., particle swarm optimization) make use of the information of the personal best of an individual and/or the entire population's best. In our framework, (7) can be utilized to identify the personal best of an individual and the entire population's best in the first phase, and (8) can serve a similar purpose in the second phase. In addition, according to the discussion in Section VI-D, generating offspring around the

current individuals could also be beneficial to the performance when using other evolutionary algorithm paradigms.

VII. CONCLUSION

In this paper, a two-phase differential evolution (ToPDE) has been proposed for uniform designs in constrained experimental domains. The main task of the first phase is to obtain a predefined number of samples in the constrained experimental domain. For this purpose, a clustering DE has been presented which divides the population into several subpopulations and guides the subpopulations toward the constrained experimental domain from different directions quickly. In addition, the second phase aims at uniformly distributing the feasible samples obtained in the first phase. To accomplish this, a new replacement strategy has been introduced. It is noteworthy that we have also designed two fitness functions for these two phases, respectively. Many experiments have been implemented in this paper. First, we studied the principle of ToPDE by a simple 2-D example. Afterward, ToPDE was applied to the uniform designs of a new type of automotive crash box and five benchmark test problems. The performance of ToPDE was compared with that of NTMD, a state-of-the-art method. The experimental results verify the effectiveness of ToPDE.

In this paper, the distance between any two samples in the population is measured by the Euclidean distance. However, the distance indicator depends largely on the type of a uniform design in a constrained experimental domain. Fig. 15 gives an example. In this example, suppose that the geodesic distance is more appropriate than the Euclidean distance. Under this condition, the Euclidean distance will severely underestimate the actual distance between two samples A and B. In the future, we will investigate other types of distance indicator in the real-world applications. Moreover, ToPDE will be applied to more uniform designs in constrained experimental domains of other fields.

The MATLAB source code of ToPDE can be downloaded from Y. Wang's homepage: <http://ist.csu.edu.cn/YongWang.htm>.

APPENDIX

The implementation of NTMD [33] for uniform designs in the constrained experimental domains defined by one linear equality constraint and several inequality constraints.

It is necessary to point out that in [33] NTMD has two variants. One is the one-pass exchange algorithm and the other is

the power-modulo-a-prime algorithm. In this paper, the one-pass exchange algorithm is employed. Moreover, we also make a simple improvement on the setting of N^* , which gradually increases during the iteration. It is because for a highly constrained problem, NTMD cannot find any feasible solution with small and moderate values of N^* . In NTMD, the linear equality constraint is formulated as follows:

$$\sum_{i=1}^D x_i = 1, A_i \leq x_i \leq B_i \quad (13)$$

where A_i and B_i are the lower and upper bounds of x_i , respectively, and $0 \leq A_i \leq B_i \leq 1$.

NTMD includes the following steps.

Step 1: Given a positive integer N^* , let $\vec{h} = (h_1, h_2, \dots, h_{D-1})$ be a vector satisfying the following conditions: a) $h_1 < h_2 < \dots < h_{D-1} < N^*$ and b) $\forall j \in \{1, 2, \dots, D-1\}$, $\gcd(N^*, h_j) = 1$, where $\gcd(N^*, h_j)$ denotes the greatest common divisor of N^* and h_j . The second condition means that h_j is relatively prime N^* .

Step 2: $c_{i,j} = \text{mod}(i \times h_j, N^*)$, $i = 1, 2, \dots, N^*$, $j = 1, 2, \dots, D-1$, where $\text{mod}(i \times h_j, N^*)$ denotes the remainder after dividing $i \times h_j$ by N^* . If $c_{i,j} = 0$, then reset it to N^* . The matrix consisting of $c_{i,j}$ is called an $N^* \times (D-1)$ lattice-point matrix of integers.

Step 3: $d_{i,j} = (2c_{i,j} - 1)/2N^*$, $i = 1, 2, \dots, N^*$, $j = 1, 2, \dots, D-1$. Clearly, $0 < d_{i,j} < 1$ and $\vec{d}_i = (d_{i,1}, d_{i,2}, \dots, d_{i,D-1})$ is a point in the $(D-1)$ -dimensional unit hypercube.

Step 4: Let $\vec{d}_i = (0, d_{i,1}, d_{i,2}, \dots, d_{i,D-1})$, $i = 1, 2, \dots, N^*$. For each \vec{d}_i , the following equations are calculated:

$$\Delta_D = 1 \quad (14)$$

$$\Delta_k = 1 - (d_{i,k} + d_{i,k+1} + \dots + d_{i,D}), \quad k = 2, 3, \dots, D-1 \quad (15)$$

$$e_k = \max\left(\frac{A_k}{\Delta_k}, 1 - \frac{B_1 + B_2 + \dots + B_{k-1}}{\Delta_k}\right) \quad (16)$$

$$\phi_k = \max\left(\frac{B_k}{\Delta_k}, 1 - \frac{A_1 + A_2 + \dots + A_{k-1}}{\Delta_k}\right) \quad (17)$$

$$x_{i,k} = G(d_{i,k}, e_k, \phi_k, \Delta_k, k-1) = \Delta_k \left\{ 1 - \left[d_{i,k}(1 - \phi_k)^{k-1} + (1 - d_{i,k})(1 - e_k)^{k-1} \right]^{1/(k-1)} \right\} \quad (18)$$

$$x_{i,1} = 1 - (x_{i,2} + x_{i,3} + \dots + x_{i,D}). \quad (19)$$

After the above process, \vec{d}_i is transformed into \vec{x}_i which can satisfy the linear equality constraint.

Step 5: Store \vec{x}_i ($i = 1, 2, \dots, N^*$) into a predefined set if \vec{x}_i satisfies all inequality constraints. If the number of points in the set is larger than or equal to NP' , i.e., a predefined number of the sample size, then stop; otherwise add the value of N^* and go to step 1.

REFERENCES

- [1] M. Anghileri, L.-M. L. Castelletti, F. Invernizzi, and M. Mascheroni, "A survey of numerical models for hail impact analysis using explicit finite element codes," *Int. J. Impact Eng.*, vol. 31, no. 8, pp. 929–944, 2005.
- [2] F. S. Almeida and A. M. Awruch, "Design optimization of composite laminated structures using genetic algorithms and finite element analysis," *Compos. Struct.*, vol. 88, no. 3, pp. 443–454, 2009.
- [3] G. G. Wang and S. Shan, "Review of metamodeling techniques in support of engineering design optimization," *ASME J. Mech. Design*, vol. 129, no. 4, pp. 370–380, 2007.
- [4] R. J. Yang, N. Wang, C. H. Tho, J. P. Bobineau, and B. P. Wang, "Metamodeling development for vehicle frontal impact simulation," *ASME J. Mech. Design*, vol. 127, no. 5, pp. 1014–1020, 2005.
- [5] S. Shetty and L. Nilsson, "Multiobjective reliability-based and robust design optimisation for crashworthiness of a vehicle side impact," *Int. J. Veh. Design*, vol. 67, no. 4, pp. 347–367, 2015.
- [6] T. Ray, M. Asafuddoula, H. K. Singh, and K. Alam, "An approach to identify six sigma robust solutions of multi/many-objective engineering design optimization problems," *ASME J. Mech. Design*, vol. 137, no. 5, 2015, Art. no. 051404.
- [7] L. Gu, "A comparison of polynomial based regression models in vehicle safety analysis," in *Proc. ASME Design Eng. Tech. Conf. Design Autom. Conf.*, Pittsburgh, PA, USA, 2001, pp. 509–514.
- [8] J. Fang, G. Sun, N. Qiu, N. H. Kim, and Q. Li, "On design optimization for structural crashworthiness and its state of the art," *Struct. Multidisciplinary Optim.*, Sep. 2016, doi: 10.1007/s00158-016-1579-y.
- [9] S. Shan and G. G. Wang, "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions," *Struct. Multidisciplinary Optim.*, vol. 41, no. 2, pp. 219–241, 2010.
- [10] A. I. J. Forrester and A. J. Keane, "Recent advances in surrogate-based optimization," *Progr. Aerosp. Sci.*, vol. 45, nos. 1–3, pp. 50–79, 2009.
- [11] E. Stinstra, D. den Hertog, P. Stehouwer, and A. Vestjens, "Constrained maximin designs for computer experiments," *Technometrics*, vol. 45, no. 4, pp. 340–346, 2003.
- [12] D. C. Montgomery, *Design and Analysis of Experiments*, 8th ed. Hoboken, NJ, USA: Wiley, 2012.
- [13] K.-T. Fang and D. K. J. Lin, "Uniform experimental designs and their applications in industry," *Handbook Stat.*, vol. 22, pp. 131–170, 2003.
- [14] T. W. Simpson, D. K. J. Lin, and W. Chen, "Sampling strategies for computer experiments: Design and analysis," *Int. J. Rel. Appl.*, vol. 2, no. 3, pp. 209–240, 2001.
- [15] A. A. Giunta, S. F. Wojtkiewicz, and M. S. Eldred, "Overview of modern design of experiments methods for computational simulations," in *Proc. 41st AIAA Aerosp. Sci. Meeting Exhibit*, Reno, NV, USA, 2003, Art no. AIAA-2003-0649.
- [16] A. B. Owen, "Orthogonal arrays for computer experiments, integration and visualization," *Statistica Sinica*, vol. 2, no. 2, pp. 439–452, 1992.
- [17] A. S. Hedayat, N. J. A. Sloane, and J. Stufken, *Orthogonal Arrays: Theory and Applications*. Springer, 2012.
- [18] L. Franek and X. Jiang, "Orthogonal design of experiments for parameter learning in image segmentation," *Signal Process.*, vol. 93, no. 6, pp. 1694–1704, 2013.
- [19] J.-S. Park, "Optimal Latin-hypercube designs for computer experiments," *J. Stat. Plan. Inference*, vol. 39, no. 1, pp. 95–111, 1994.
- [20] H. Dette and A. Pepelyshev, "Generalized Latin hypercube design for computer experiments," *Technometrics*, vol. 52, no. 4, pp. 421–429, 2012.
- [21] B. G. M. Husslage, G. Rennen, E. R. van Dam, and D. D. Hertog, "Space-filling Latin hypercube designs for computer experiments," *Optim. Eng.*, vol. 12, no. 4, pp. 611–630, 2011.
- [22] R. Li, "Model selection for analysis of uniform design and computer experiment," *Int. J. Rel. Qual. Safety Eng.*, vol. 9, no. 4, pp. 367–382, 2002.
- [23] X. Li, X. Li, and Y. Su, "A hybrid approach combining uniform design and support vector machine to probabilistic tunnel stability assessment," *Struct. Safety*, vol. 61, pp. 22–42, Jul. 2016.
- [24] Y. Kuo, T. Yang, B. A. Peters, and I. Chang, "Simulation metamodel development using uniform design and neural networks for automated material handling systems in semiconductor wafer fabrication," *Simulat. Model. Pract. Theory*, vol. 15, no. 8, pp. 1002–1015, 2007.
- [25] K.-T. Fang, D. K. J. Lin, P. Winker, and Y. Zhang, "Uniform design: Theory and application," *Technometrics*, vol. 42, no. 3, pp. 237–248, 2000.

- [26] S. Golchi and J. L. Loepky, "Monte Carlo based designs for constrained domains," *arXiv:1512.07328v2*, 2016. [Online]. Available: <https://arxiv.org/abs/1512.07328>
- [27] L. Duan *et al.*, "Crashworthiness design of vehicle structure with tailor rolled blank," *Struct. Multidisciplinary Optim.*, vol. 53, no. 2, pp. 321–338, 2016.
- [28] C. H. Chuang, R. J. Yang, G. Li, K. Mallela, and P. Pothuraju, "Multidisciplinary design optimization on vehicle tailor rolled blank design," *Struct. Multidisciplinary Optim.*, vol. 35, no. 6, pp. 551–560, 2008.
- [29] Z.-F. Qi, J.-F. Yang, Y. Liu, and M.-Q. Liu, "Construction of nearly uniform designs on irregular regions," *Commun. Stat. Theory Methods*, 2016. [Online]. Available: <http://dx.doi.org/10.1080/03610926.2016.1179755>
- [30] D. Draguljić, T. J. Santner, and A. M. Dean, "Noncollapsing space-filling designs for bounded nonrectangular regions," *Technometrics*, vol. 54, no. 2, pp. 169–178, 2012.
- [31] Y.-Z. Liang, K.-T. Fang, and Q.-S. Xu, "Uniform design and its applications in chemistry and chemical engineering," *Chemometr. Intell. Lab. Syst.*, vol. 58, no. 1, pp. 43–57, 2001.
- [32] L. Zhang, Y.-Z. Liang, J.-H. Jiang, R.-Q. Yu, and K.-T. Fang, "Uniform design applied to nonlinear multivariate calibration by ANN," *Analytica Chimica Acta*, vol. 370, no. 1, pp. 65–77, 1998.
- [33] J. J. Borkowski and G. F. Piepel, "Uniform designs for highly constrained mixture experiments," *J. Qual. Technol.*, vol. 41, no. 1, pp. 35–47, 2009.
- [34] Y. Liu and M.-Q. Liu, "Construction of uniform designs for mixture experiments with complex constraints," *Commun. Stat. Theory Methods*, vol. 45, no. 8, pp. 2172–2180, 2016.
- [35] K.-T. Fang and Z.-H. Yang, "On uniform design of experiments with restricted mixtures and generation of uniform distribution on some domains," *Stat. Probab. Lett.*, vol. 46, no. 2, pp. 113–120, 2000.
- [36] R.-B. Chen, Y.-W. Hsu, Y. Hung, and W. Wang, "Discrete particle swarm optimization for constructing uniform design on irregular regions," *Comput. Stat. Data Anal.*, vol. 72, no. 3, pp. 282–297, 2014.
- [37] R. Lekivetz and B. Jones, "Fast flexible space-filling designs for non-rectangular regions," *Qual. Rel. Eng. Int.*, vol. 31, no. 5, pp. 829–837, 2015.
- [38] E. Benková, R. Harman, and W. G. Müller, "Privacy sets for constrained space-filling," *J. Stat. Plan. Inference*, vol. 171, pp. 1–9, Apr. 2016.
- [39] 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.
- [40] Y. Wang and Z. Cai, "A dynamic hybrid framework for constrained evolutionary optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 1, pp. 203–217, Feb. 2012.
- [41] Y. Shi, P. Zhu, L. Shen, and Z. Lin, "Lightweight design of automotive front side rails with TWB concept," *Thin Walled Struct.*, vol. 45, no. 1, pp. 8–14, 2007.
- [42] S. J. Lee *et al.*, "Design flow for the crash box in a vehicle to maximize energy absorption," *Proc. Inst. Mech. Eng. D J. Automobile Eng.*, vol. 227, no. 2, pp. 179–200, 2014.
- [43] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
- [44] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.
- [45] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 55–66, Feb. 2011.
- [46] Y. Wang, B.-C. Wang, H.-X. Li, and G. G. Yen, "Incorporating objective function information into the feasibility rule for constrained evolutionary optimization," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2938–2952, Dec. 2016.
- [47] Y. Wang, H.-X. Li, G. G. Yen, and W. Song, "MOMMOP: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 830–843, Apr. 2015.
- [48] V. Santucci, M. Baiocchi, and A. Milani, "Algebraic differential evolution algorithm for the permutation flowshop scheduling problem with total flowtime criterion," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 682–694, Oct. 2016.
- [49] S. C. Chuang and Y. C. Hung, "Uniform design over general input domains with applications to target region estimation in computer experiments," *Comput. Stat. Data Anal.*, vol. 54, no. 1, pp. 219–232, 2010.
- [50] J. J. Liang *et al.*, "Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization," Nanyang Technol. Univ., Singapore, Tech. Rep., Sep. 2006.
- [51] W. M. Duckworth, II, "Some binary maximin distance designs," *J. Stat. Plan. Inference*, vol. 88, no. 1, pp. 149–170, 2000.
- [52] M. E. Johnson, L. M. Moore, and D. Ylvisaker, "Minimax and maximin distance designs," *J. Stat. Plan. Inference*, vol. 26, no. 2, pp. 131–148, 1990.



Yong Wang (M'08) received the B.S. degree in automation from the Wuhan Institute of Technology, Wuhan, China, in 2003, and the M.S. degree in pattern recognition and intelligent systems and the Ph.D. degree in control science and engineering from Central South University (CSU), Changsha, China, in 2006 and 2011, respectively.

He is currently an Associate Professor with the School of Information Science and Engineering, CSU. His research interests include theory, algorithm design, and applications of computational

intelligence.

Dr. Wang was a recipient of the Hong Kong Scholar by the Mainland—Hong Kong Joint Post-Doctoral Fellows Program, China, in 2013, the Excellent Doctoral Dissertation by Hunan Province, China, in 2013, the New Century Excellent Talents in University by the Ministry of Education, China, in 2013, the 2015 IEEE Computational Intelligence Society Outstanding Ph.D. Dissertation Award, the Hunan Provincial Natural Science Fund for Distinguished Young Scholars in 2016, and the EU Horizon 2020 Marie Skłodowska-Curie Fellowship in 2016. He is currently serving as an Associate Editor for *Swarm and Evolutionary Computation*. He was a Reviewer of 50+ international journals and a PC Member of 40+ international conferences. He is a member of the IEEE CIS Task Force on Nature-Inspired Constrained Optimization and the IEEE CIS Task Force on Differential Evolution.



Biao Xu received the B.S. degree in automation from the Hunan Institute of Technology, Hengyang, China, in 2013, and the M.S. degree in control engineering from Central South University, Changsha, China, in 2016.

He is currently an Algorithm Engineer in Huawei, Shenzhen, China. His research interests include algorithm design and applications of telecommunication and computational intelligence.



Guangyong Sun received the B.S. and Ph.D. degrees in mechanical engineering from Hunan University, Changsha, China, in 2003 and 2011, respectively.

He is currently an ARC Discovery Early Career Researcher with the School of Aerospace, Mechanical and Mechatronic Engineering, Faculty of Engineering, University of Sydney, Sydney, NSW, Australia. His research interests include lightweight design, structural optimization, and automotive safety.



Shengxiang Yang (M'00–SM'14) received the B.Sc. and M.Sc. degrees in automatic control and the Ph.D. degree in systems engineering from Northeastern University, Shenyang, China, in 1993, 1996, and 1999, respectively.

He is currently a Professor of Computational Intelligence and Director with the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, U.K. He has over 220 publications. His research interests include evolutionary computation, swarm intelligence, computational intelligence in dynamic and uncertain environments, artificial neural networks for scheduling, and relevant real-world applications.

Dr. Yang serves as an Associate Editor or Editorial Board Member of eight international journals, including the IEEE TRANSACTIONS ON CYBERNETICS, *Information Sciences*, *Evolutionary Computation*, *Neurocomputing*, and *Soft Computing*.