Soft Comput (2016) 20:3023–3044
DOI 10.1007/s00500-015-1929-5

FOCUS

Application of permutation genetic algorithm for sequential model building–model validation design of experiments

Mohammed Reza Kianifar[1] · Felician Campean[1] · Alastair Wood[1]

Published online: 9 December 2015

Abstract The work presented in this paper is motivated by a complex multivariate engineering problem associated with engine mapping experiments, which require efficient design of experiments (DoE) strategies to minimise expensive testing. The paper describes the development and evaluation of a Permutation Genetic Algorithm (PermGA) to enable an exploration-based sequential DoE strategy for complex real-life engineering problems. A known PermGA was implemented to generate uniform OLH DoEs, and substantially extended to support generation of model building–model validation (MB–MV) sequences, by generating optimal infill sets of test points as OLH DoEs that preserve good space-filling and projection properties for the merged MB + MV test plan. The algorithm was further extended to address issues with non-orthogonal design spaces, which is a common problem in engineering applications. The effectiveness of the PermGA algorithm for the MB–MV OLH DoE sequence was evaluated through a theoretical benchmark problem based on the Six-Hump-Camel-Back function, as well as the Gasoline Direct Injection engine steady-state engine mapping problem that motivated this research. The case studies show that the algorithm is effective in delivering quasi-orthogonal space-filling DoEs with good properties even after several MB–MV iterations, while the improvement in model adequacy and accuracy can be monitored by the engineering analyst. The practical importance of this work, demonstrated through the engine case study, is that significant reduction in the effort and cost of testing can be achieved.

Keywords Design of experiments · Optimal Latin hypercube · Permutation genetic algorithm · Model-based engine calibration

1 Introduction

The driving factor for the work presented in this paper stems from an engineering problem associated with efficient engine test planning for calibration development. The ever increasing complexity of engine technologies towards improving performance, fuel economy, and drivability while meeting increasingly stringent emissions legislation has resulted in a complex and time-consuming process of calibrating the controllable engine parameters (e.g. fuel rail pressure and start of injection). To address this challenge, model-based calibration strategies have been utilised (Roepke 2009; Kruse et al. 2010), commonly underpinned by statistical methodologies for planning physical engine tests (on engine dynamometers) and developing behavioural models for an engine based on response surface methodology. Specifically, steady-state engine mapping is based on statistical modelling of the engine responses of interest using test data collected at fixed speed/load operating points (Roepke 2009). The choice of location of test points in the design space associated with the actuators ranges at each engine operating point has a key role in ensuring the adequacy and accuracy of engine response models, as the basis for subsequent calibration optimisation studies (Kruse et al. 2010). Design of Experiments strategies have been adopted for engine test planning. In general, an efficient DoE strategy aims to minimise the cost of testing

Communicated by D. Neagu.

✉ Mohammed Reza Kianifar
m.kianifar@bradford.ac.uk

Felician Campean
F.Campean@bradford.ac.uk

Alastair Wood
A.S.Wood@bradford.ac.uk

[1] School of Engineering, University of Bradford, Bradford, UK

while maximising the information content, such that response surface models of specified approximation accuracy can be developed (Bates et al. 2003).

The practical importance of choosing an efficient DoE strategy is associated with the high cost of engine testing. Given that modern engine calibration problems involve an increasing number of calibration variables (often more than 10), conventional (factorial-based) DoE strategies are generally not feasible or economical. A further complication of engine experiments is that the design space is often not orthogonal with respect to one or more variables (i.e. linear or nonlinear constraints limit the actuator space in one or more dimensions), which further limits the applicability of classical DoEs. The DoE methods commonly used for engine mapping experiments include D-Optimal and V-Optimal DoEs and space-filling DoEs (Seabrook et al. 2003; Grove et al. 2004; Sacks et al. 1989; McKay et al. 2000). Space-filling DoEs, in particular those based on optimal Latin hypercubes (OLH), have been increasingly used for engine model-based calibration problems, given that they enable more flexible models to describe the engine behaviour over a wider design space (e.g. 'global' models over the whole engine speed-load space), with no prior knowledge required regarding the type of model that would adequately represent the trends (Seabrook et al. 2005). The OLH DoEs have the advantage that the number of test points can be set by the analyst based on experience and resource limitations. However, this raises the risk of test plans that generate an insufficient amount of information due to *under-sampling* with the implication that the required model accuracy is not achieved. Conversely, if a larger OLH DoE test plan is selected, this raises the possibility of *over-sampling*, wasting time and energy by collecting more tests than needed.

Recent research work in fields dealing with similar testing cost issues (e.g. electronics, chemistry, and aerodynamics) has focused on the development of sequential DoE approaches that iteratively augment an initial DoE with further test points until the desired model quality is reached (Crombecq et al. 2011; Geest et al. 1999; Provost et al. 1999). This strategy can facilitate a higher testing efficiency compared to the fixed size tests commonly used in practice, and has the advantage that it can flexibly adapt to modelling complexity requirements of different engine responses. In general, sequential DoEs can be divided into two main categories:

1. *Optimal sequential design*: For this type of sequential DoE, the model type and its parameters are known in advance (e.g. polynomial). This allows the algorithms to use the behaviour of the set model type to guide the sampling points into the right direction within the design space; e.g. the D-optimal designs minimise the covariance of the model parameters estimates (Draguljić et al. 2012). The main issue with these DoEs is that if the assumed model type is not suitable for the response, the DoE plan is not efficient and the enhancement in model accuracy through collecting more data is not guaranteed.
2. *Evolutionary sequential design*: Given that the type of model may not be known in advance for many engineering problems, and therefore a nonparametric model is required, justifies the need for a generic sequential DoE that makes no assumptions about the model type, number of sample points or system behaviour. Such DoEs use the information from previous iterations to decide where to select the next test point (Crombecq et al. 2009). These evolutionary sequential DoEs can be further classified into:

   (i) *Exploitation-based sequential design methods* (Geest et al. 1999; Forrester et al. 2008): Exploitation-based DoEs use an error measure from the previous steps to guide the sampling points to the interesting parts of design space, e.g. areas with discontinuous system behaviour or areas containing optima. The main problem with exploitation-based DoEs is the tendency to over-focus on specific areas, which could leave some part of design space under-sampled.

   (ii) *Exploration-based sequential design methods* (Provost et al. 1999; Crombecq and Dhaene 2010; Crombecq et al. 2009, 2011): Exploration-based sequential DoEs give equal importance to all regions of design space and aim to fill it up as evenly as possible at each sequence. In this method, the location of the test points from the previous iteration is used as feedback for sampling new test points, ensuring that not too many or too few samples are collected from the same regions of design space. These DoEs are not specifically linked with any response models and aim to distribute the points evenly through the design space.

Considering the fact that the engine calibration is a complex nonlinear multivariate engineering problem with high level of uncertainty associated with the behaviour of responses, an optimal sequential DoE will not always be a useful DoE option given that knowledge or specification of the model type is required in advance. Additionally, an exploitation-based sequential DoE may result in early dismissal of potential calibration solutions given that some parts of the high dimensional design space could be left unexplored.

The approach proposed by the authors (Kianifar et al. 2013, 2014) is to use an exploration-based sequential DoE strategy based on optimal space-filling designs (OLH DoEs), deployed as a model building–model validation (MB–MV) DoE sequence (Narayanan et al. 2007). A key feature of the approach is that each DoE (i.e. both MB and MV) in the sequence is an OLH DoE, however, the merged MB–MV DoE, while it is optimised for space fillingness, it does not strictly follow the Latin hypercube rule, being instead made up of interlaced levels in the individual OLH DoEs. This addresses the limitation of the approach proposed by Narayanan et al. (2007) which requires the number of MV iterations and the number of tests in each MV design to be known a priori. It also addresses the limitations of other sequential DoE strategies used in engine mapping problems based on Sobol Sequences (Lam 2008), which are not adaptive with respect to 'learning' from previous stages and the selection of new test points is quasi-random from any subset of design space where the discrepancy is low.

Generating OLH designs as individual designs or as a subset of larger designs to have an exploration-based optimisation strategy is a complex optimisation problem since the aim is to preserve the whole system space fillingness within subsequent sequences of OLH designs. It is not practical to build an optimal Latin hypercube (OLH) design through enumeration, since considering all possible combinations of variables is expensive and time-consuming. For example, for a simple problem of 10 sample points and 5 variables, there are 6×10^{32} possible combinations. If each solution takes one nanosecond (1×10^{-9} s) to evaluate, the whole evaluation process would take approximately 2×10^{16} years (Fuerle and Sienz 2011), which is clearly impractical. Different global optimisation algorithms have been proposed in the literature for generating OLH designs, such as column wise–pairwise, simulated annealing, and Permutation Genetic Algorithm (PermGA) (Bates et al. 2003, 2004; Liefvendahl and Stocki 2006; Audze and Eglais 1977).

Genetic optimisation algorithm is a population-based stochastic search method inspired from genes behaviour, and it is one of the most robust random search methods due to the element of directed search (Shukla and Deb 2007). GA has been broadly used as an alternative to the classical optimisation algorithms for solving complex engineering optimisation problems (Bertram 2014; Dhingra et al. 2014; Deb et al. 2014). PermGA is working based on the same principles as the standard GA algorithm, however, the PermGA's optimisation operators (e.g. crossover and mutation) are modified to work with permuted numbers to solve discrete optimisation problems, as discussed by Bates (Bates et al. 2003). However, to support the proposed exploration-based sequential DoE strategy, the PermGA algorithm needs to be further developed, and its performance the evaluated in relation to the type of engineering problems that have motivated this work—which is the aim of the work presented in this paper.

The paper first outlines the statistical requirements needed to design an efficient exploration-based sequential DoE strategy, and then describes in detail the proposed MB–MV DoE approach, including the choice space-filling metrics. The implementation of the proposed DoE strategy using a modified PermGA algorithm for generalised infill OLH DoEs is presented next, illustrated with simple theoretical examples. The proposed approach of PermGA-exploration-based sequential OLH DoE is then validated theoretically through application to on a mathematical test-case, and empirically through application to an industrial problem of steady-state mapping of a gasoline direct injection (GDI) engine. The paper ends with a discussion of the results and opportunities for further work.

2 Problem definition

Applying a sequential exploration-based DoE strategy has the potential to improve the testing methodology by achieving the model target accuracy through less data points, particularly when the testing process is time-consuming and expensive. However, efficiency of a sequential space-filling DoE strategy is highly dependent on the quality of the design augmentation technique to fulfil several statistical requirements. Accordingly, to design an efficient space-filling augmentation strategy there is a need to consider four important criteria:

(i) *Non-collapsingness*: A non-collapsing design [i.e. with good projective properties (Van Dam et al. 2007)] guarantees that no two sample points project onto each other along any of the axes when the K-dimensional sample points are projected into the $(K - 1)$-dimensional space. In other words, in a non-collapsing design each sample point has a unique value along any of the axes (Van Dam et al. 2007). In effect, the projection criterion ensures that every parameter is represented over its domain, even if the response is only dominated by a few of the parameters.

(ii) *Granularity*: Granularity is an important requirement for sequential designs (Crombecq et al. 2011). Granularity indicates the proficiency of the DoE strategy to augment the initial experimental design by small batches of additional test points. Accordingly, a fine-grained sequential strategy is flexible regarding the total size of DoE samples, despite the number of design variables and levels, which consequently results in avoiding over- or under-sampling (Hartmann and Nelles 2013; Klein et al. 2013).

(iii) *Space fillingness*: This is the fundamental principle for an exploration-based sequential DoE technique, which requires to distribute the sample points (i.e. collect information) evenly within the design space regardless of the problem dimension and sample size (Van Dam et al. 2007; Ye et al. 2000; Joseph and Hung 2008; Morris and Mitchell 1995; Johnson et al. 1990).

(iv) *Orthogonality*: This criterion ensures that there is no correlation between any combination of input parameters (Tang 1993; Owen 1992), thus ensuring that the experimental design is a good representative of the real variability (Khan 2011). Taking into the consideration that only a few existing experimental designs are orthogonal (e.g. factorial designs) most of the existing space-filling strategies attempt to reasonably satisfy the orthogonality criterion.

In this research, a model building–model validation (MB–MV) sequential DoE strategy is proposed to efficiently fulfil the four statistical requirements discussed above.

3 Model building–model validation DoE framework

3.1 MB–MV DoE strategy

The strategy adopted in this research uses optimal Latin hypercube (OLH) space-filling DoEs (Fuerle and Sienz 2011; Bates et al. 2003; Liefvendahl and Stocki 2006) as the basis for both model building (MB) and model validation (MV) DoEs. Within the proposed algorithm, additional infill test points (generated as OLH DoE) are iteratively added to an initial model building OLH DoE, until the required modelling accuracy is achieved. At each iteration, the additional infill points generated are treated as an external validation set, used to evaluate the model quality. If modelling accuracy is not satisfactory, the MB and MV OLH DoEs are merged into a new model building set and a further MV set is collected for the next iteration.

The proposed strategy provides a good fit with the practical requirements of engineering problems such as the steady-state engine testing problem. By using the MB–MV strategy, a smaller MB OLH DoE experiment can be planned (e.g. $m = 50$ test points), followed by a validation (MV) DoE experiment (e.g. $v = 15$ test points). The MV is also an OLH but the optimality criterion is to minimise the space-filling metric across the union of the MB and MV sets ($m + v = 65$ test points). Engine response models are fitted based on the MB DoE data, typically, using non-parametric or semi-parametric models [such as Kriging or Radial Basis Functions (RBF)], and the quality of the models is evaluated via the prediction error (e.g. using the root mean squared error, RMSE) for the validation set, i.e. the MV DoE test data (Narayanan et al. 2007). If the model accuracy requirements are not met, a further validation DoE test (MV2) is planned with v_2 test points, using the same principle of a OLH MV DoE in which the space-filling metric is minimised across the combined MB + MV + MV2 set. A new model is fitted using the MB + MV set, and validated against the MV2 set. This process is repeated iteratively until the model accuracy requirement is met. The flowchart of the approach is illustrated in Fig. 1.

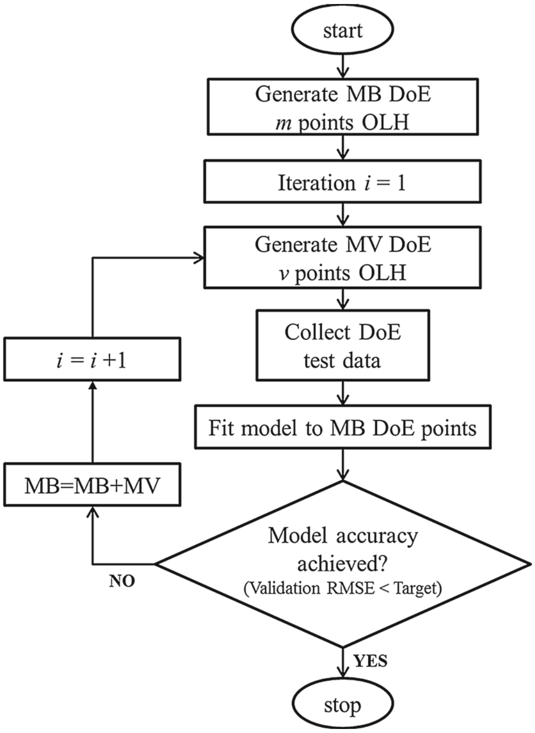


Fig. 1 MB–MV strategy flowchart

The computational challenge is to develop an efficient algorithm to support the implementation of the proposed MB–MV strategy, while satisfying the four requirements to have an efficient sequential DoE strategy (i.e. non-collapsingness, granularity, space fillingness, and orthogonality).

3.2 MB–MV DoE implementation

The proposed MB–MV DoE strategy uses the Latin hypercube (LH) principle. A LH design is generated by gridding the design space of each parameter into N (i.e. sample size) equidistant levels, and selecting only one test point on each level. Therefore, a LH design ensures that all levels of each parameter are represented over its range by maintaining non-collapsingness (Sacks et al. 1989). A LH design can be defined as:

$$L = \begin{pmatrix} x_1^1 & x_1^2 & \dots & x_1^K \\ x_2^1 & x_2^2 & \dots & x_2^K \\ \vdots & \vdots & \vdots & \vdots \\ x_N^1 & x_N^2 & \dots & x_N^K \end{pmatrix}$$

L is a LH design where K denotes the number of dimensions. In this matrix, each row represents a design point while each column shows the design points in one dimension. LH-based DoEs are popular space-filling DoE techniques due to their unique ability to generate *non-collapsing* designs, which is essential in ensuring uniformity of space exploration in all dimensions (Van Dam et al. 2007).

Unlike the other sequential DoE strategies based on OLH designs, such as the sequential nested Latin hypercube DoE method (Crombecq et al. 2011), the MB–MV design is fine-grained. In this design, the number of additional MV points at each sequence is arbitrary, e.g. small batches of OLH test points, whereas the nested LH design doubles the number of test points at each iteration.

To maintain a LH design with good *space-filling* properties, the optimal Latin hypercube (OLH) DoEs are generated by minimising a chosen metric for space filling or uniformity metric. The distribution of the test points for an OLH design is regarded as a discrete optimisation problem (Bates et al. 2004).

The main challenges with this optimisation problem are:

(i) Formulation of the optimisation objective function to maintain space fillingness.
(ii) Development of an effective algorithm for the discrete optimisation problem.

3.2.1 Optimisation objective function

Several uniformity metrics for OLH DoEs have been described in literature. Table 1 gives an overview of the frequently employed optimality criteria to generate an OLH design with a good space-filling property.

Of the optimality criteria shown in Table 1, both Maximin (Van Dam et al. 2007; Ye et al. 2000; Joseph and Hung 2008; Morris and Mitchell 1995; Johnson et al. 1990) (i.e. maximising the minimum distance between every two samples) and Audze and Eglais (1977) functions have been proven to maintain a good inter-site distance. Maximin criteria tend to generate more sample points around the corners, especially for high dimensional problems (Draguljić et al. 2012). Consequently, this strategy might not preserve good space-filling properties at the centre of the design space, particularly for a small sample size (N). Draguljić et al. (2012) discussed that the Audze–Eglais criterion performs better for high dimensional problems (K).

Table 1 Summary of common space-filling criteria

Optimality criterion	Formula
Manhattan (Van Dam et al. 2007; Ye et al. 2000)	$\min_{\mathbf{x_i},\mathbf{x_j} \in N} \sum_{k=1}^{K} \lvert x_i^k - x_j^k \rvert$
Maximin (Van Dam et al. 2007; Ye et al. 2000; Joseph and Hung 2008; Morris and Mitchell 1995; Johnson et al. 1990)	$\min_{\mathbf{x_i},\mathbf{x_j} \in N} \sqrt{\sum_{k=1}^{K} \lvert x_i^k - x_j^k \rvert^2}$
Audze–Eglais (AELH) (Audze and Eglais 1977)	$\sum_{\mathbf{x_i},\mathbf{x_j} \in N} \sqrt{\sum_{k=1}^{K} \lvert x_i^k - x_j^k \rvert^2}$
$\emptyset_p$ (Morris and Mitchell 1995)	$\left(\sum_{\mathbf{x_i},\mathbf{x_j} \in N} \sqrt{\sum_{k=1}^{K} \lvert x_i^k - x_j^k \rvert^2}^{\,p} \right)^{1/p}$

The requirements for the MB–MV DoE strategy are to (i) generate OLH MB DoEs with good space-filling properties; (ii) generate an infill set of validation points as an OLH DoE that would project good space-filling properties in conjunction with the initial DoE; and (iii) the algorithm must also be robust to generate an infill set of validation points within non-orthogonal variables design spaces.

On this basis, the uniformity metric chosen for the OLH DoEs is the Audze–Eglais Latin hypercube (AELH) potential energy concept, given its superior robustness in dealing with variable dimensionality and sample size (Bates et al. 2004). The AELH function considers the test points as mass units where each mass unit exerts repulsive force to the other units until it reaches the equilibrium position (i.e. minimum potential energy). In this function the magnitude of the repulsive forces is inversely proportional to the square of the distance between the mass units (i.e. points) in the system. Thus, the AELH objective function can be presented as follows (Bates et al. 2003, 2004):

$$U = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \frac{1}{L_{ij}^2} \tag{1}$$

U is the potential energy, N denotes the total number of DoE points, and L_{ij} is the distance between any two points i and $j, i \neq j$. Minimising the potential energy ensures a uniform distribution of the sample points within the design space.

To generate an optimal sequence of DoEs according to the MB–MV principle, it is proposed that the space-filling criteria is maintained throughout the MB–MV sequence. What this means is that MB and MV DoEs are generated as an OLH, but the optimality criterion for the MV DoE is defined in relation to the space-filling metric for the overall DoE sequence (i.e. including all MB and MV DoE test points). The joint DoE created by the union of the MB DoE (based on N levels) and MV DoE (based on M levels) test points will not strictly fulfil the LH principle as it is based on $N + M$ interlaced and, in general, unevenly distributed levels. Therefore, for the optimal augmentation of the DoE, i.e. by generating optimal ‘infill’ points for the MV set, the same uniformity metric (i.e. the AELH function) is used. The main challenge for this step is to modify the AELH objective function to take into account the position of the MB points already fixed in the design space. This means that the fitness function should include both the M new test points and the existing N test points from the MB OLH DoE. The fitness function is modified accordingly, shown in Eq. (2).

$$U = \left(\sum_{i=1}^{M} \sum_{j=i+1}^{M} \frac{1}{L_{ij}^2} + \sum_{i=1}^{M} \sum_{n=1}^{N} \frac{1}{L_{in}^2} \right) \tag{2}$$

U is the potential energy, L_{ij} is the distance between any two points i and j, $(i \neq j)$ in the MV OLH DoE, and L_{in} is the distance between each new point i and the exiting points n. The outcome of the MB–MV design step will be an OLH design with an optimal uniform distribution of points across the design space, with the new MV points optimally filling the under-sampled areas in the original design.

Equations (1) and (2) are capable of generating the OLH test points within a symmetric design space in each of the dimensions. For many engineering problems the design space might be severely constrained in relation to some design variables (i.e. asymmetric design space). This could impair the ability of the search algorithm to generate enough valid points for the MB, or any of the subsequent MVs, and affect the space-filling quality of the generated design (Fuerle and Sienz 2011).

A variety of constraint-handling methods for evolutionary algorithms have been proposed, as summarised by Michalewicz and Schoenauer (1996), and Mezura-Montes and Coello Coello (2011). The commonly used strategies are:

1. *Repair* strategy, The idea of this strategy is that an infeasible individual is *repaired* to a feasible individual (Liepins and Vose 1990).
2. *Sudden Death* strategy, in which an infeasible individual is *removed* immediately from the population (Schwefel 1993).
3. *Penalty functions*, the basic idea of this strategy is to refine the fitness functions by extending the objective function with a penalty term. Penalty functions are the most commonly used approaches for evolutionary algorithm, in particular for handling inequality constraints (Barbosa et al. 2015).

In this work, the latter strategy was adopted by implementing the sequential unconstrained minimization technique (SUMT) to generate OLH designs with constrained design spaces (Byrne 2012). This technique is based on adding an increasing penalty function to the objective function to avoid unnecessary computational costs by generating test points that are not feasible or do not have a physical meaning. Accordingly, the objective function in Eq. (2) was modified as shown in Eq. (3).

$$U = \left(\left[\sum_{i=1}^{M} \sum_{j=i+1}^{M} \frac{1}{L_{ij}^2} + c_t \times P(x) \right] + \left[\sum_{i=1}^{M} \sum_{n=1}^{N} \frac{1}{L_{in}^2} \right] \right) \tag{3}$$

$$P(x) = \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{G} (\max\{0, g_j(x_i)\})^2 \tag{4}$$

In Eq. (3), c_t denotes the monotonically increasing penalty parameter: $c_{t+1} = \eta \times c_t$ where $\eta > 1$. Therefore, the penalty parameter is increasing iteratively during the PermGA process until all the infeasible points are directed into the feasible area. Also, $P(x)$ is a function of the inequality design constraints ($g_j(x) < 0$, where $j = 1, \ldots, G$), as given in Eq. (4). Noteworthy, all the G constraints in Eq. (4) are scaled, to ensure that the penalty term generated by each constraint is of the same magnitude.

3.2.2 *PermGA optimisation algorithm*

The distribution of the test points for an OLH design can be regarded as a discrete optimisation problem (Bates et al. 2004). Table 2 summarises some examples of the optimality criteria and optimisation algorithms employed to develop an OLH design.

It has been argued in the literature that PermGA can be more efficient for higher dimensional OLH DoE problems due to a convergence rate corresponding to the varying number of variables (Bates et al. 2004; Liefvendahl and Stocki 2006). In other words, PermGA can generate better distributed points for high dimensional DoEs while reducing the computational costs.

Therefore, a PermGA algorithm was developed and implemented in this paper to generate the MB–MV designs. Another reason to select the PermGA is that it is a population-based stochastic optimiser; thus, it is expected that the algorithm generates a number of random permuted populations for each dimension with no particular correlation among the generated points. So, it is expected that the final DoE generated by PermGA preserves good orthogonality properties.

Table 2 Examples of the introduced optimisation techniques in literature to construct an OLH design

Reference	Optimisation algorithm
Audze and Eglais (1977)	Coordinates exchange algorithm
Morris and Mitchell (1995)	Simulated annealing
Ye et al. (2000)	Column-wise–pairwise
Bates et al. (2003)	PermGA
Bates et al. (2004)	PermGA/Simulated annealing
Liefvendahl and Stocki (2006)	PermGA/Column-wise–pairwise
Van Dam et al. (2007)	Branch-and-bound algorithm

I. PermGA Development

The pseudocode for the PermGA algorithm implemented in Matlab environment is provided in Algorithm 1.

```
Algorithm 1. PermGA
begin
i = 0
1. Generate a number of random LH designs (i.e. based on GA
    population size) using permuted numbers
2. While the termination conditions are not met do
  begin
  3. Evaluate the individual's fitness function (i.e. for MB
      DoE using equation (1), for MV DoE with symmetric
      design space of variables using equation (2), and for MV
      DoE with asymmetric design space of variables using
      equation (3))
  4. Store a number of individuals with the best fitness values
      (i.e. elites)
  5. While the number of new LH designs (i.e. children) is less
      than the set population size do
      begin
      6. Select two of the individuals (i.e. parents)
      7. Apply Cycle and Inversion crossover operators
          considering the crossover rate
      8. Apply mutation operator considering the mutation rate
      end
  i = i +1
  Update penalty parameter (c) if the design space of variables
  is asymmetric
  end
end
```

To enhance the exposition of developed PermGA algorithm, the extended design structure matrix (XDSM), which is an extension of a common diagram in system engineering (Lambe and Martins 2012), was employed to visualise the interconnections among the PermGA components, as shown in Fig. 2.

Following the XDSM convention for architectural decomposition (Lambe and Martins 2012), PermGA algorithm components are represented by rectangles, special components which control the iterations (known as drivers) are shown by rounded rectangles, and data are represented by parallelograms. The function of components is to process data. The thick grey lines are used to show the data flow, while the thin black lines illustrate the process flow. The input data transfers to the components from the vertical direction and departs the components from the horizontal direction. The convention for the data flow is that connections above the diagonal flow from left to right and top to bottom, and connections below the diagonal flow from right to left and bottom to top. Accordingly, parallelograms at the column above and below the components define the input data, and parallelograms along the row define the output data. Moreover, external inputs and outputs are placed on the outer edges of the diagram, in the top row and leftmost column, respectively (Lambe and Martins 2012). Another XDSM architecture convention is that any block referring to component *i* represents a repeated pattern. In addition, a numbering system is used to illustrate the order of components execution, i.e. it starts from zero and proceeds in numerical order. In this numbering system, the loops are shown by $j \to k$ for $k < j$, which denotes that the algorithm returns to step *k* until the required termination condition by the driver is met. For further details, see Lambe and Martins (2012).

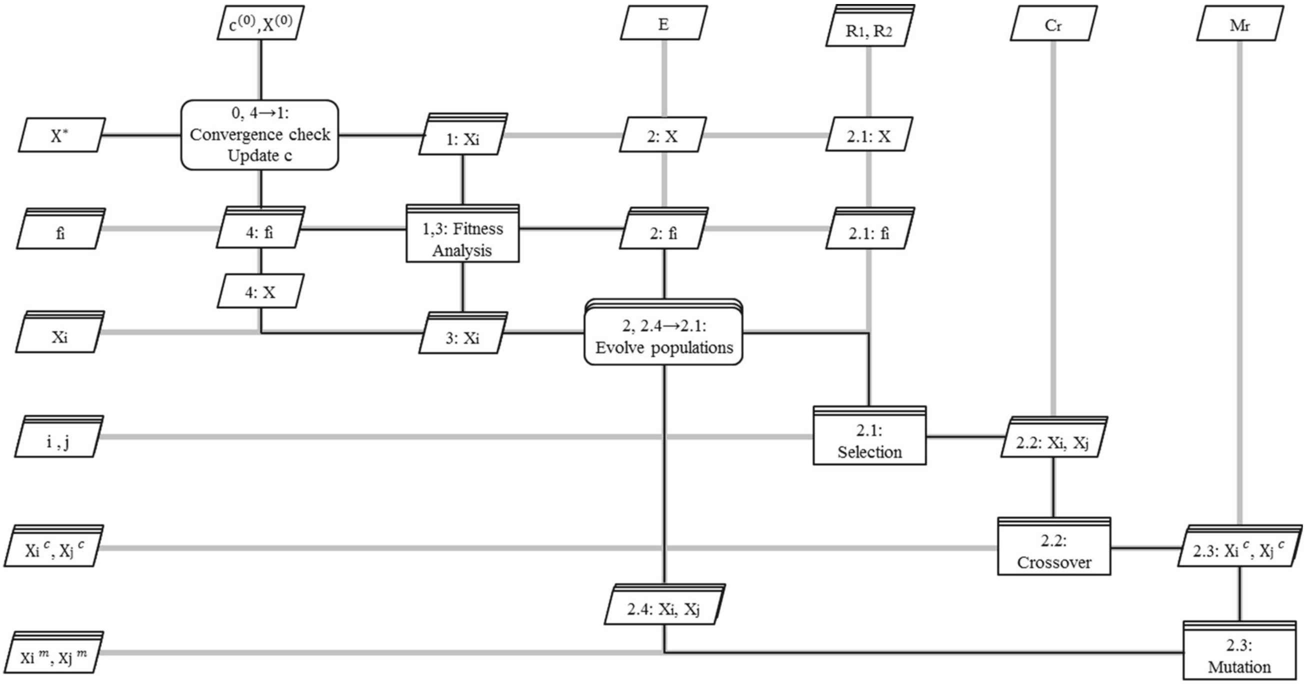


Fig. 2 Illustration of Permutation GA algorithm process using the XDSM graph

For implementation, the external inputs at **level 0** (shown as **X**) are N populations of RLH designs which have been generated using the ‘Permutation’ encoding (Michalewicz 1996) in Matlab. Also, if the design space of variables is constrained, an initial value for penalty parameter (c) is another external input at this level. At **level 1**, the optimisation objective function is calculated for each input population $\mathbf{X}_i$ ($i = 1, \ldots, N$). This iterative analysis component is shown as ‘Fitness Analysis’ in Fig. 2. The fitness function for the MB points is calculated using Eq. (2) for MV points within a symmetric design space, and Eq. (3) for MV points within an asymmetric design space (i.e. constrained design spaces). Given that each DoE parameter might have a different range of units, the variables have each been normalised to the interval [0 1] to calculate the L_{pq}. Next, the fitness value for each population $\mathbf{f}_i$ is transferred to **level 2** (shown as ‘Evolve populations’), along with the initial populations **X**. In this level, GA operators were applied to the initial population (parents) to evolve the new population (children). In this implementation, ‘selection’, ‘crossover’ and ‘mutation’ GA operators are utilised, shown as **level 2.1**, **level 2.2**, and **level 2.3**, respectively, in Fig. 2.

- **‘Selection’ operator (Level 2.1)**: This operator defines the method of selecting the parent populations to be evolved. Several selection methods have been discussed in literature, including ‘Tournament’ and ‘Biased Roulette Wheel’ (Coley 1999). In this work, the ‘Biased Roulette Wheel’ operator was implemented to increase the convergence rate, by giving individuals with better fitness values $\mathbf{f}_i$ proportionally more chance to be selected as parents. R_1 and R_2 external inputs are two random numbers which define the parent populations ($\mathbf{X}_i$ and $\mathbf{X}_j$).
- **‘Crossover’ operator (Level 2.2)**: This operator combines parts of input parent populations (X_i and X_j) and generates two new individuals (X_i^c and X_j^c). There are different crossover methods used for PermGA, such as: simple crossover, cycle crossover and inversion (Bates et al. 2004). Bates et al. (2004) have shown that either cycle crossover or inversion works well for a PermGA algorithm. However, given that the interactions among GA parameters are complex and dependent on the fitness function (Deb and Agrawal 1999), it was decided to employ both crossover functions, i.e. cycle crossover followed by inversion. The cycle crossover preserves the absolute position of the elements in the parent sequence (Fig. 3), while using inversion crossover the points are inverted between two sets of points (Narayanan et al. 2007) (Fig. 4). The aim was to introduce extra variability into the children populations to reduce the chance of the search algorithm being trapped in a local optima.

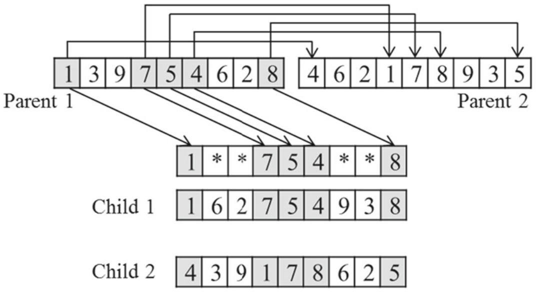


Fig. 3 Cycle crossover

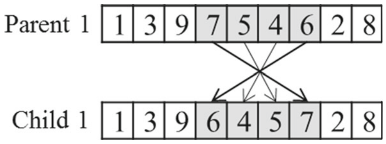


Fig. 4 Inversion crossover

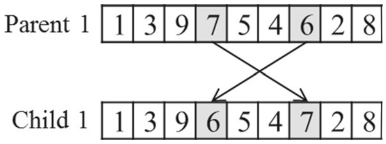


Fig. 5 Mutation operator

- **‘Mutation’ operator (Level 2.3)**: A simple mutation technique (Liefvendahl and Stocki 2006; Michalewicz 1996) was used to swap two randomly selected elements of the transferred individuals, from the crossover level (X_i^c and X_j^c), and evolve them into new child populations (X_i^m and X_j^m), as shown in Fig. 5.

The output of the iterative process at level 2 is N new populations which are transferred to level 3 to evaluate the fitness of evolved populations. Finally, the convergence requirements are checked at level 4. If the convergence requirements are met, the optimum solution (X^*), which is the final OLH design, is delivered. Otherwise, the new population, along with the updated penalty parameter (c), if the design space is asymmetric, will be transferred to level 1 for another iteration of program.

Additionally, some PermGA parameters, shown as external inputs in Fig. 2, require tuning due to their significant influence on the algorithm performance (Grefenstette 1986); specifically:

- **Elite Size** (**E**): defines how many individuals with the best fitness value $\mathrm{f_i}$ should transfer to the next iteration of the algorithm without evolving. Thus, the best individuals are not lost during subsequent generations, which can accordingly assure a smoother convergence.

- **Crossover Rate** (**Cr**): determines the number of individuals that are evolved through the crossover operation.
- **Mutation Rate** (**Mr**): determines the number of individuals that are evolved through the mutation operation.
- **Population Number** (**N**): denotes the population size for the input design X.

II. PermGA Preliminary Results

For illustration, Fig. 6 shows the result of a MB DoE sequence for a two-dimensional problem, with 60 DoE points generated using Eq. (1), where x_1 and $x_2 \in [-1\ 1]$.

Figure 7 illustrates the space-filling properties of the MB DoE, in terms of the minimum *Euclidean* distance from the nearest point (Morris and Mitchell 1995), for each of the test points. The Euclidean distance for each sample point is calculated with Eq. (5) (Crombecq et al. 2011):

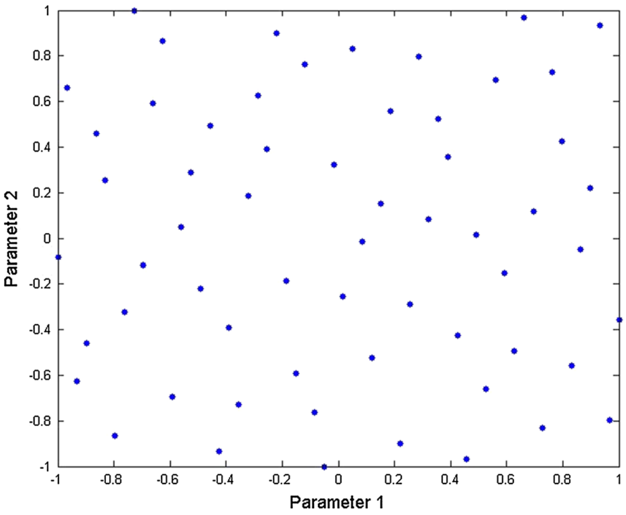


Fig. 6 MB Sequence (60 test points)

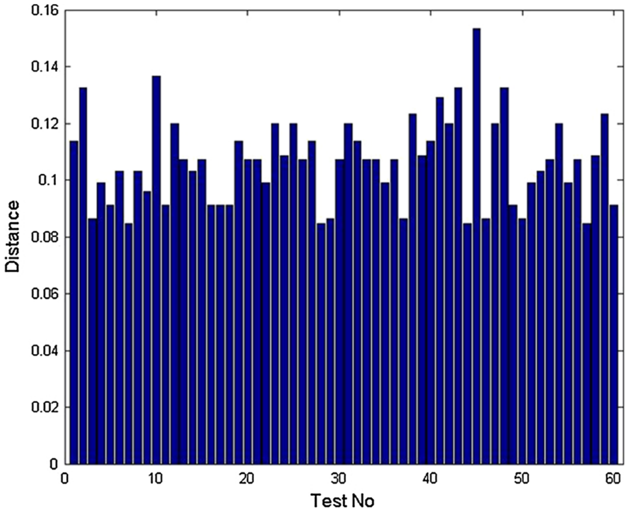


Fig. 7 Euclidean minimum distance for all MB DoE points

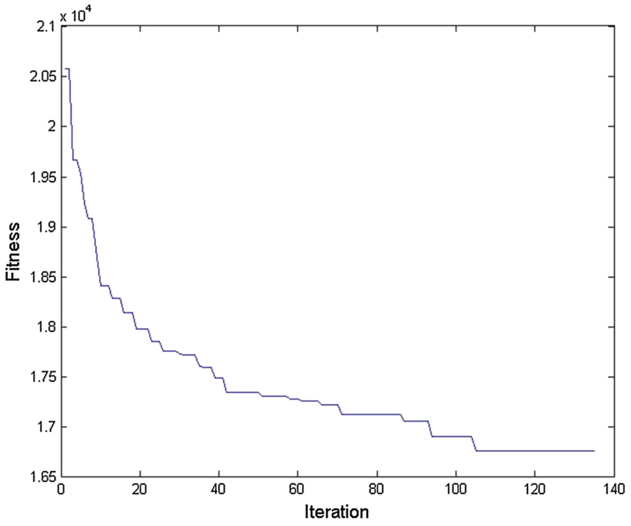


Fig. 8 Convergence plot of PermGA for generating the MB OLH DoE sequence (60 points in 2 dimensions)

$$D_i = \underset{\mathbf{x_i},\mathbf{x_j} \in N}{\text{minimum}} \sqrt{\sum_{k=1}^{K} |x_i^k - x_j^k|^2} \quad \text{where } i \neq j \tag{5}$$

N is the sample size and K denotes the number of design parameters. The graph in Fig. 7 shows a uniform distribution of the minimum distance to the nearest test point, with a mean of 0.106 and standard deviation of 0.015.

To study the orthogonality properties of the designs generated by PermGA, the correlation (r) between the vectors of variables were calculated using Eq. (6) (Joseph and Hung 2008):

$$r = \frac{\sum_i^N (x_i^k - \overline{x^k})(x_i^j - \overline{x^j})}{\sqrt{\sum_i^N (x_i^k - \overline{x^k})^2}\sqrt{\sum_i^N (x_i^j - \overline{x^j})^2}} \quad \text{where } k \neq j \tag{6}$$

where x_i^k indicates the sample i for the design parameter k and $\overline{x^k}$ gives the samples' average for the design parameter k. The correlation between vectors of variables for 60 OLH DoE points is −0.019, which means that the MB OLH design is quasi-orthogonal.

Figure 8 illustrates a typical convergence plot in terms of fitness for the MB algorithm. The PermGA algorithm was run with the following GA settings: 'Population Size' = 200; 'Crossover Rate' = 0.8, 'Mutation Rate' = 0.05, and 'Elite Size' = 5.

To illustrate the MB–MV DoE algorithm using the modified objective functions, i.e. Eqs. (2) and (3), an MV DoE of 40 points was generated using the MB DoE shown in Fig. 6. Several boundary conditions have been considered to reflect common practical engineering situations, as illustrated in the examples discussed below.

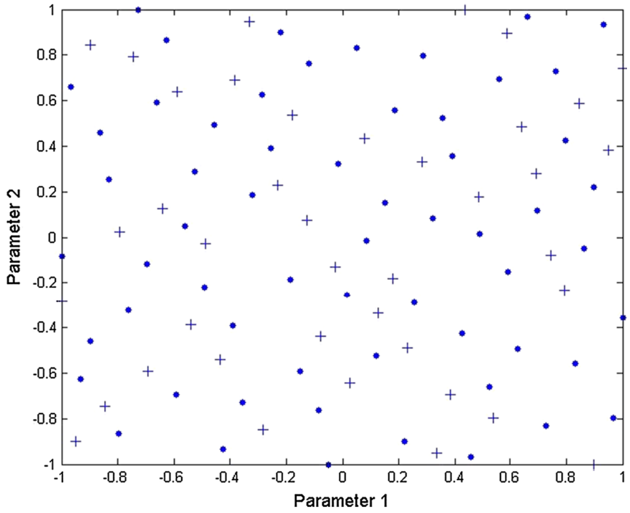


Fig. 9 MB–MV Sequence for symmetric design space

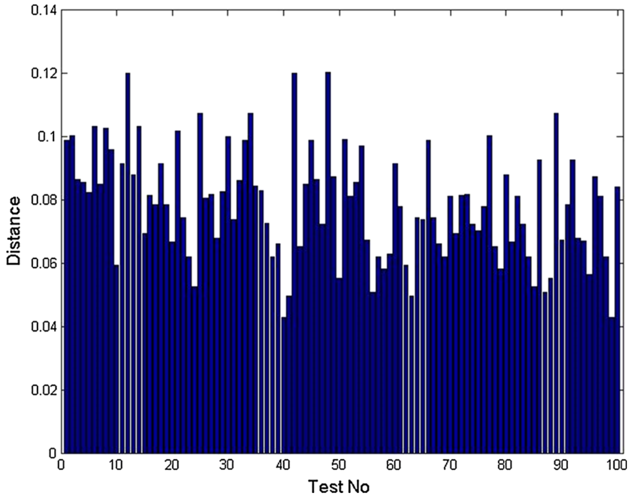


Fig. 10 Euclidean minimum distance for MB–MV test points

Example 1 40 MV DoE test points with no boundary constraint, where x_1 and $x_2 \in [-1\ 1]$.

Given that no boundary conditions are imposed, Eq. (2) was used as the objective function. The PermGA algorithm settings used were: 'Population Size' = 200; 'Crossover Rate' = 0.8, 'Mutation Rate' = 0.05, and 'Elite Size' = 5. Figure 9 illustrates the location of 40 MV infill DoE test points (cross-shaped) among the existing 60 MB points (circle dots). Similar statistical analysis was executed on the MV DoE to investigate the design space fillingness and orthogonality. Therefore, Fig. 10 illustrates the Euclidean distance for 100 DoE points (60 MB + 40 MV points), calculated using Eq. (5). This graph shows a uniform distribution of Euclidean distances, with a mean of 0.079 and standard deviation of 0.017. Also, the correlation between the vectors of x_1 and x_2 parameters for 100 DoE points is 0.025, calculated using Eq. (6), which means that the final design is still quasi-orthogonal.

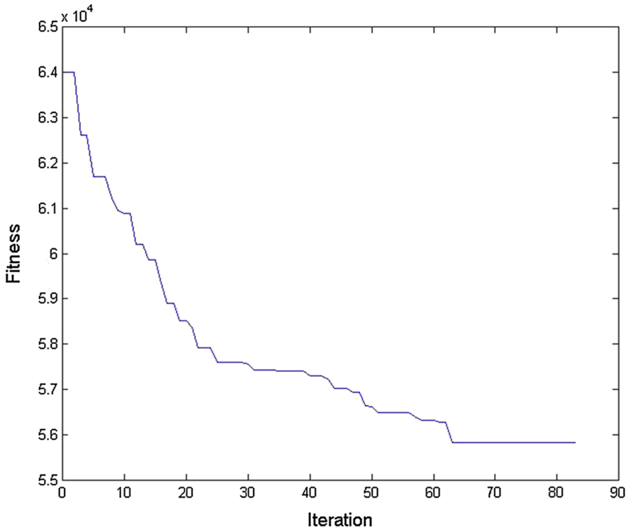


Fig. 11 Convergence plot for PermGA

Figure 11 illustrates the convergence plot in terms of fitness for the augmented MB–MV algorithm, showing that the augmented search algorithm is smoothly converged to the optimum solution.

Example 2 40 MV DoE test points with no boundary constraint, where x_1 and $x_2 \in [0\ 1]$.

For many real-life engineering problems, it is the case that after the first step of characterizing the response model using the initial DoE, there is a need to collect more data from a smaller part of the design space to achieve a better model accuracy. One of the advantages of using an efficient sequential DoE strategy is that the design space can be modified sequentially, which in effect provides the opportunity to revise the design space after data analysis at each DoE sequence using the same DoE principals. Therefore, in this example, the design space for the two parameters at the MV stage was revised from [−1 1] to [0 1]. The PermGA algorithm was run with 'Population Size' 200; 'Crossover Rate' 0.8, 'Mutation Rate' 0.05, and 'Elite size' 5.

Figure 12 illustrates the location of 40 MV DoE test points (cross-shaped) with revised design space, generated using Eq. (2) as the objective function, among the existing 60 MB points (circle dots). The square box in Fig. 12 shows the revised design space, within which the MV test points are generated. Figure 13 shows the Euclidean distances for all the 100 DoE points, with a mean of 0.149 and standard deviation of 0.062. This figure shows that the distribution of Euclidean distances (or space fillingness) is not as smooth as Example 1, i.e. higher standard deviation for Euclidean distances, which was expected since the design space of the additional 40 infill

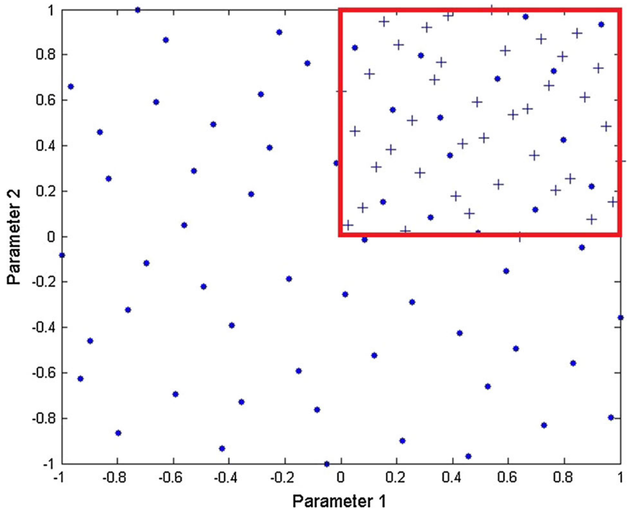


Fig. 12 MB–MV Sequence for revised design space

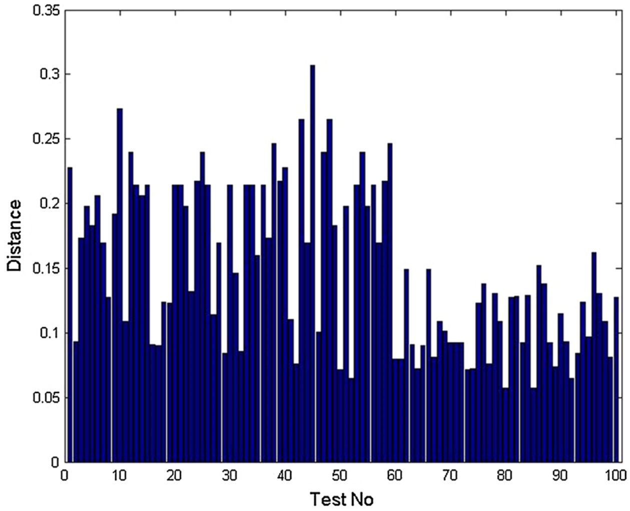


Fig. 13 Euclidean minimum distance for MB–MV test points

points was not equal to the design space of the 60 MB test points.

The correlation between vectors of x_1 and x_2 parameters for the total design of 100 test points is 0.18, which is still an acceptable value for the correlation term, i.e. within the [−0.3 0.3] limit suggested by Steinberg and Lin (2006). Figure 14 illustrates the convergence plot in terms of fitness for the augmented MB–MV algorithm, showing that the augmented search algorithm is smoothly converged to the optimum solution.

Example 3 40 MV DoE test points with 1 boundary constraint $g(x) = x_2 - x_1 - 1 \leq 0$, where x_1 and $x_2 \in [-1\ 1]$.

In this example, an asymmetric design space was considered, i.e. linearly constrained by the inequality constraint $g(x) = x_2 - x_1 \leq 1$. The augmented infill PermGA algorithm, with the fitness function given in Eq. (3), including an

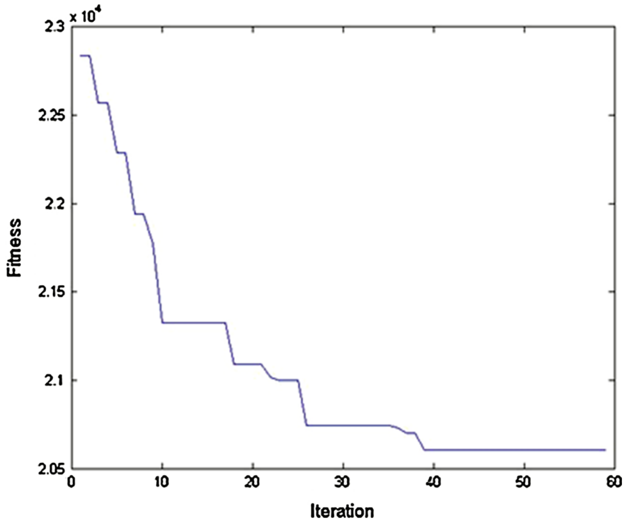


Fig. 14 Convergence plot for PermGA

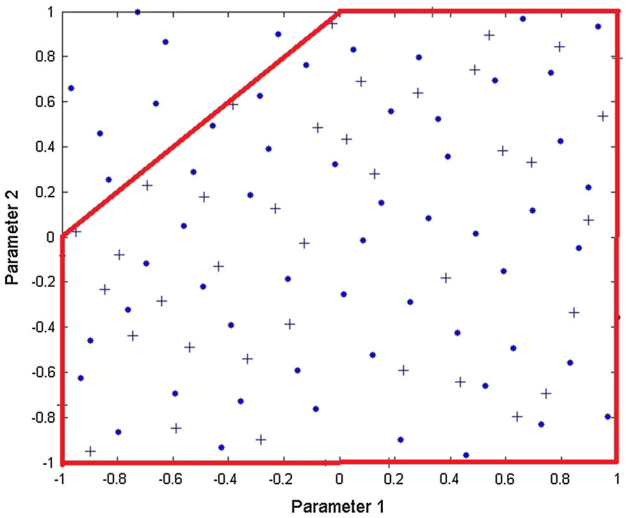


Fig. 15 MB–MV Sequence for asymmetric design space

adaptive penalty function was used to generate the 40 infill MV test points, as illustrated in Fig. 15. The PermGA algorithm was run with ‘Population Size’ 200; ‘Crossover Rate’ 0.8, ‘Mutation Rate’ 0.05, and ‘Elite size’ 5.

In Fig. 15, the red lines show the feasible design space for the MV DoE sequence. The space-filling uniformity for the MB–MV OLH DoE points in the constrained design space is illustrated in Fig. 16, based on the *Euclidean* distance to the nearest test point. This figure shows that the infill points are distributed within the constrained design space, with a mean of 0.074 and standard deviation of 0.018. The correlation between vectors of x_1 and x_2 parameters for the total design of 100 test points is 0.13, which is an acceptable value for the correlation term.

Figure 17 illustrates the convergence plot in terms of fitness for the augmented MB–MV algorithm, showing that

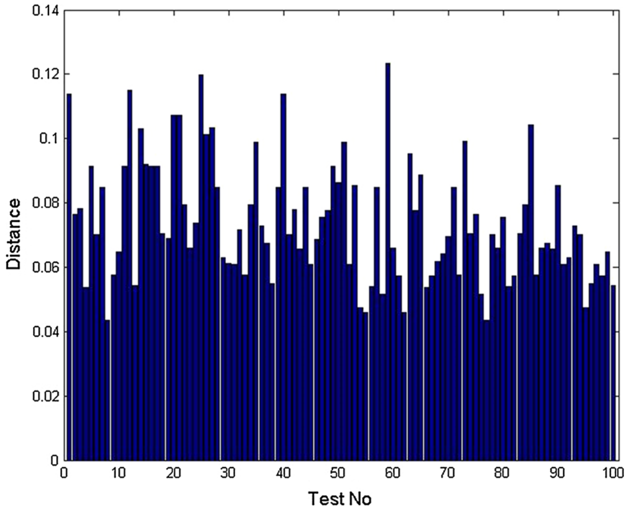


Fig. 16 Euclidean minimum distance for MB–MV test points

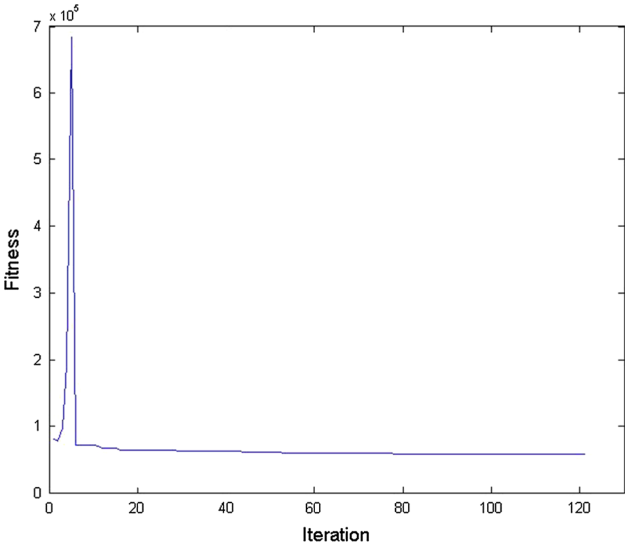


Fig. 17 Convergence plot for PermGA

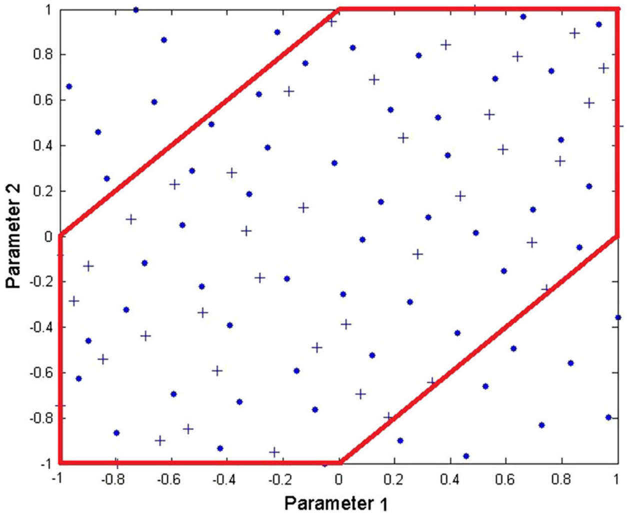


Fig. 18 MB–MV Sequence for asymmetric design space

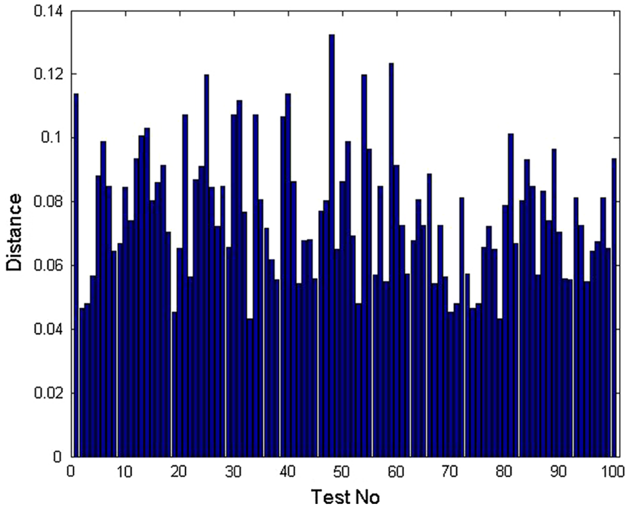


Fig. 19 Euclidean minimum distance for MB–MV test points

the augmented search algorithm is converged to the optimum solution. The figure shows that the fitness value has increased for 6 iterations as the GA population with the best fitness value (i.e. the solution of the GA process at each iteration) could not meet the inequality constraint. In effect, the increase in the fitness value was due to the sequentially increasing penalty term. Then, from the 6th iteration onwards the best population was within the linearly constrained design space; thus, the fitness value was decreased significantly since no penalty term was applied.

Example 4 40 MV DoE test points with 2 boundary constraints, $g_1(x) = x_2 - x_1 - 1 \leq 0$ and $g_2(x) = x_1 - x_2 - 1 \leq 0$, where x_1 and $x_2 \in [-1\ 1]$.

In this example, the design space is linearly constrained by two inequality constraints $g_1(x) = x_2 - x_1 \leq 1$ and $g_2(x) = x_1 - x_2 \leq 1$. Figure 18 illustrates the 40 infill MV test points (cross-shaped), generated using Eq. (3) as the fitness function, among the 60 MB test points (circle dots). In this figure, the red lines show the feasible design space for the MV DoE sequence. The PermGA algorithm was run with ‘Population Size’ 200; ‘Crossover Rate’ 0.8, ‘Mutation Rate’ 0.05, and ‘Elite size’ 5.

The space-filling uniformity for the MB–MV OLH DoE points in the constrained design space is illustrated in Fig. 19, based on the *Euclidean* distance, distributed with a mean of 0.077 and standard deviation of 0.02. Also, the correlation between vectors of x_1 and x_2 parameters for the total design of 100 test points is 0.22, which is still an acceptable value for the correlation term.

Figure 20 illustrates the convergence plot in terms of fitness for the augmented MB–MV algorithm. Similar to the

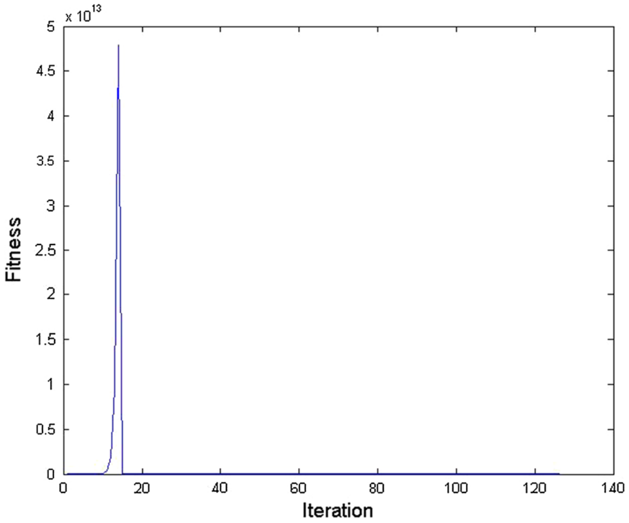


Fig. 20 Convergence plot for PermGA

previous example, this figure shows an exponential increase in the fitness value for the initial iterations until iteration 16, until a feasible solution within the design space constraints was found.

4 Validation case studies

To validate the application of the MB–MV DoE strategy implemented through the developed PermGA algorithm, two case studies were considered, illustrating both theoretical and empirical (via a real-world engineering case study) validation of the approach.

4.1 Theoretical validation via benchmark problem: the six hump camel back (SHCB) function

The SHCB function, given in Eq. (7), is a surrogate engineering problem which is a well-known example for evaluating the global optimisation plans (Wang et al. 2004). This function has a complex shape with six local optima and two global optima of -1.0316 at $(0.0898, -0.7127)$ and $(-0.0898, 0.7127)$.

$$f(x) = 4x_1^2 - 2.1x_1^4 + 1/3x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4 \\ \text{where } x_1 \in [-2, 2], x_2 \in [-1, 1] \qquad (7)$$

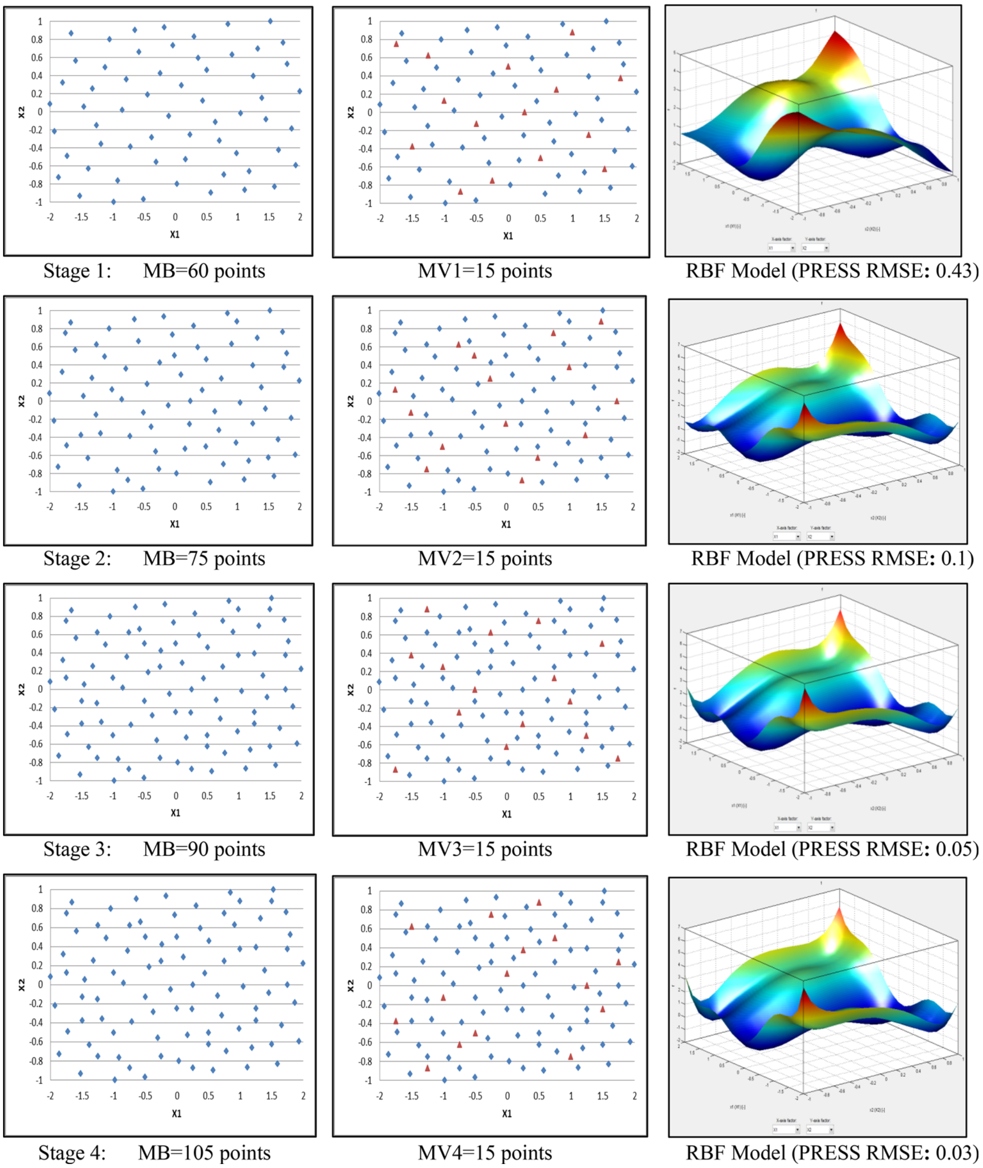
For the purpose of comparing the model accuracy after each iteration of MB–MV DoE strategy, the MB–MV DoE was planned in four iterations. In the first step, an MB OLH DoE with 60 points was generated using Eq. (1). A MV OLH DoE with 15 points was generated as the first Model Validation design (MV1) using Eq. (2). The SHCB function was evaluated at both MB and MV1 points, and a response model was fitted based on the MB test points in the MATLAB MBC Toolbox™, using Radial Basis Functions (RBF) models (Fang and Mark 2005; Forrester et al. 2008) [i.e. RBF with Thinplate Kernel function (Morton and Knott 2002)]. In step two, the same type of RBF response model was built based on the joint MB + MV1 test points (i.e. 75 points). A second Model Validation (MV2) was generated based on a 15 points OLH MV DoE. The same process of internal and external validation (MV2 points) was applied. This process was repeated with two further iterations, with MV3 = 15 points and MV4 = 15 points (i.e. in the 4th iteration the model building set comprised MB + MV1 + MV2 + MV3 = 105 points, an MV4 = 15 points).

Figure 21 illustrates the distribution of points in the DoEs at each iteration. Figure 22 shows the uniformity of the distribution of the points in the design space, in terms of the *Euclidean* distance for each of the 120 test points in the joint MB–MV DoEs. This histogram indicates that distribution of the Euclidean distance for the test points is quasi-uniform and even after 4 independent steps of testing the distributed points are still remote from each other within the design space (i.e. test points are not replicated). For better illustration of the uniformity of the distributed test points within the design space (i.e. space-filling properties), the distributions of the minimum Euclidean distance of the test points across the subsequent MV DoEs are illustrated using boxplots, as shown in Fig. 23. This figure indicates that:

1. The *Euclidean* distance of test points decreases by adding more test points (from MB–MV1 to MB–MV4), which was expected since the number of test points within the finite design space are increasing over the MV DoE sequences.
2. The variability of the *Euclidean* distances measured via the interquartile range decreases across the subsequent DoEs. This trend demonstrates the ability of PermGA algorithm to enhance the space-filling properties of the merged DoEs, by preserving the uniform distribution of test points when generating the MV DoEs.

Figure 23 shows a number of outliers from collecting more data (i.e. 'MB–MV3' and 'MB–MV4'). The main reason for the appearance of the outliers is that the spread of the Euclidean distances is smaller for the 'MB–MV3' and 'MB–MV4' DoEs, due to a more uniform distribution of Euclidean distances. Accordingly, the interquartile range, and consequently the whiskers, is smaller in these DoEs, which in effect increases the possibility of having more test points with Euclidean distances out of the whiskers' range (i.e. outliers).

Figure 24 characterises the distribution of the Euclidean distances in terms of its standard deviation across the 4 stages. This graph shows that the uniform space-filling properties of



Stage 1: MB=60 points MV1=15 points RBF Model (PRESS RMSE: 0.43)

Stage 2: MB=75 points MV2=15 points RBF Model (PRESS RMSE: 0.1)

Stage 3: MB=90 points MV3=15 points RBF Model (PRESS RMSE: 0.05)

Stage 4: MB=105 points MV4=15 points RBF Model (PRESS RMSE: 0.03)

Fig. 21 MB–MV DoE projection and response surface modelling for SHCB problem

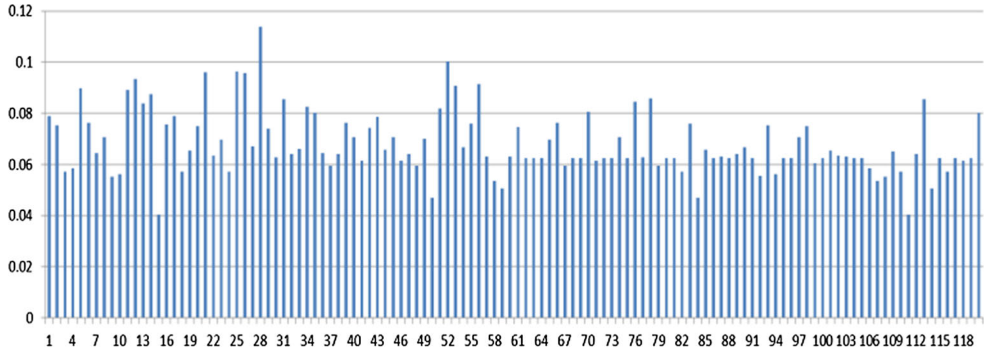


Fig. 22 *Euclidean* 'Maxi–min' (Mm) distances of all the sample points in four steps

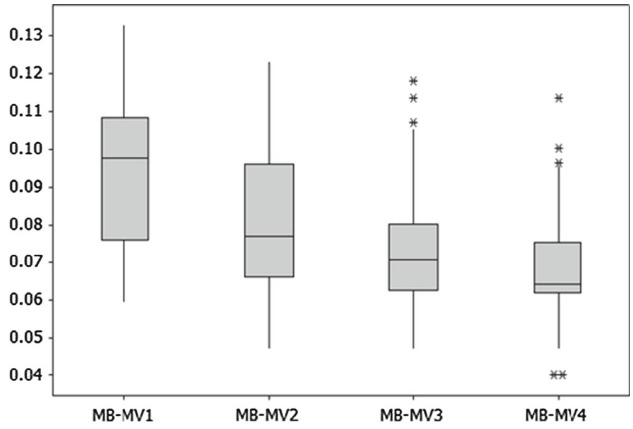


Fig. 23 *Boxplot* of *Euclidean* distances across the subsequent validation DoEs

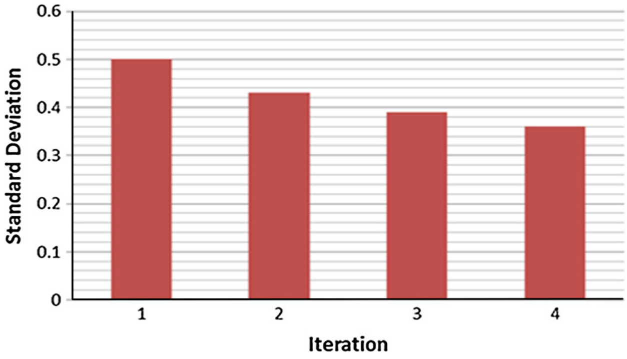


Fig. 24 Standard deviation (σ) of the Mm Distance (Stages 1 to 4)

the MB–MV DoEs are improving over the subsequent stages of DoE. The correlation between x_1 and x_2 after 4 stages was also calculated as $r = 0.05$, which indicates that the final design is quasi-orthogonal.

Furthermore, Fig. 21 gives a graphical illustration of the response surfaces fitted at each stage. These graphs clearly show that the accuracy of the model improves through the successive MB–MV stages. By looking at the internal model validation criterion (PRESS RMSE, given in Fig. 21) and external validation (RMSE for the MV test points, shown in Fig. 25), it can be concluded that the accuracy of the model improves dramatically over the first 3 stages, with only a small improvement between the 3rd and 4th stages.

The main conclusion from this study was that the proposed MB–MV sequential DoE framework is successful at generating a quasi-orthogonal DoE with good space-filling properties. The proposed design is also a fine-grained design, augmented iteratively with small batches of MV points, with good projection properties, since it uses batches of OLH designs to cover the whole range of design space for each parameter.

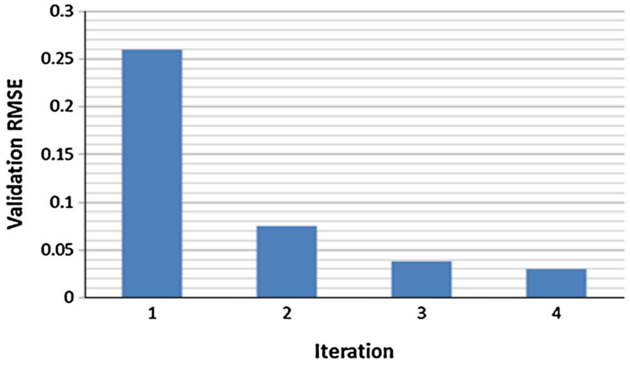


Fig. 25 External validation of the built models through MB–MV sequence (Stages 1 to 4)

4.2 Empirical validation: application to a GDI engine steady-state engine mapping

The GDI engine case study described in this paper is based on engine dynamometer experiments conducted in the powertrain testing facility at the University of Bradford for the part load "hot" steady-state calibration of a 5-l naturally aspirated V8 GDI engine. This case study was based on the model-based steady-state calibration process discussed in (Dwyer et al. 2013), with testing conducted at a number of engine speed-load operating points, to study the effect of calibration variables on fuel consumption and emissions.

The MB–MV DoE strategy, implemented through the algorithms described in this paper, was used to generate the test plan for the GDI engine mapping experiments, and to develop response models of sufficient accuracy for the calibration optimisation process. Table 3 and Fig. 26 summarise the engine calibration control variables and the engine responses of interest at each engine speed/load operating point.

From a calibration engineering viewpoint, the fuel consumption and particulates number (PN) emissions are responses that are of interest for calibration optimisation, which can be defined as identifying variables settings to minimise fuel consumption and PN emissions. However, combustion stability and exhaust gases temperature are in fact state variables, which act as nonlinear constraints for the calibration optimisation problem. From a DoE strategy point of view, collecting test points in areas where the engine

Table 3 GDI engine calibration parameters

Name	Description	Limits		Unit
IVO	Inlet Valve Opening	−12	50	deg ATDC
EVC	Exhaust Valve Closing	−6	44	deg ATDC
FRP	Fuel Rail Pressure	8	15	MPa
SOI	Start of Injection	260	335	deg BTDC

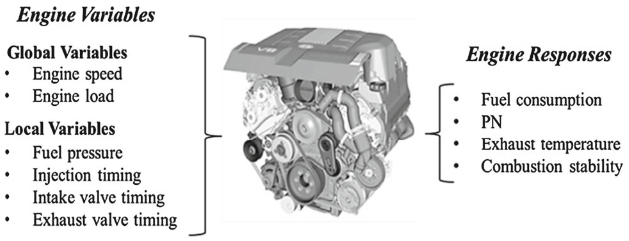


Fig. 26 GDI engine calibration parameters and responses

operation is infeasible from the point of view of these state variables would be a waste. Thus, combustion stability and exhaust temperature act as nonlinear constraints for the DoE problem.

The approach adopted for the GDI mapping case study was to design and run a preliminary screening experiment as an OLH DoE. Response models were fitted based on the screening experiment and used to define a revised variable space based on the evaluation of the combustion stability and exhaust temperature responses. For example, it was observed that for lower speed/load engine operating points, negative valve overlap (valve overlap can be defined as the time that both the inlet and exhaust vales are open, i.e. Overlap $=$ EVC $-$ IVO), results in poor combustion stability. This is consistent with engineering judgment where, at part load, high overlap results in excessive exhaust gas recirculation (EGR) with negative effects upon combustion stability, especially under low load conditions such as idle (Hagen and Holiday 1976). Based on this analysis, a negative overlap constraint was introduced for the design space for IVO and EVC calibration variables at low speed/load operating points for the MB–MV sequence.

4.2.1 MB–MV DoE strategy implementation

The implementation plan for the sequential DoE generated a sequence of MB–MV OLH DoEs using the developed PermGA algorithms. Given the constrained design space for IVO and EVC parameters at some of the minimap points, Eq. (3) was used as fitness function for the individual OLH designs during the MB–MV process.

Similar to the SHCB case study, the model building DoE was planned as an OLH DoE with 50 test points, with each subsequent model validation DoE of size 15, i.e. MV1 $= 15$ validation test points for the main MB DoE, iteratively augmented with subsequent MVs of size 15 (following the process outlined in Fig. 2), until engine response models of satisfactory quality are achieved. Within the case study, to validate the sequential DoE methodology, four MB–MV iterations were planned and run, and the performance of the models were evaluated after each iteration. The Matlab MBC toolbox was used to fit Radial Basis Function (RBF) models (Fang and Mark 2005; Forrester et al. 2008) for all engine responses; preliminary evaluation of RBF models showed that the thin-plate kernel provided good models across the engine responses of interest. The RBF model selection criterion (including the number of basis functions) was based on minimising PRESS RMSE. The quality of the model was judged based on statistical diagnostics: validation RMSE (i.e. RMSE for prediction errors of the new test data in the validation set), and PRESS RMSE for the MB set (i.e. root mean square of prediction sum of squared errors for the MB set based on simple cross-validation). The model residuals for the MB set were also monitored at each DoE stage to ensure that models are not over-fitting.

4.2.2 Engine case study results and discussion

Figure 27 illustrates the Euclidean distance of the test points at each of the model validation stages using boxplots. This figure shows that variability of the Euclidean distances decreases by adding the subsequent MV DoEs, which in effect shows the capability of the developed PermGA algorithm to distribute the points evenly within the four-dimensional design space even after 4 independent sequences of MV DoEs. Also, 4 outliers were seen at ‘MB–MV4’ DoE stage. The Euclidean distance of these outliers are not worse (less or more) than the previous DoE stage (i.e. ‘MB–MV3’), however; since the variability is less at ‘MB–MV4’ (i.e. interquartile range is smaller) these test points are recognised as outliers. Moreover, Fig. 28 characterises the distribution of Euclidean distances in terms of its standard deviation across the 4 MB–MV stages. This figure shows that the uniformity of the distribution of the test points within the design space is improving across the subsequent stages of MV DoEs.

The correlation (r) between each of the two DoE variables was also studied for all the 4 stages of the MB–MV strategy. It was observed that the correlation between the variables

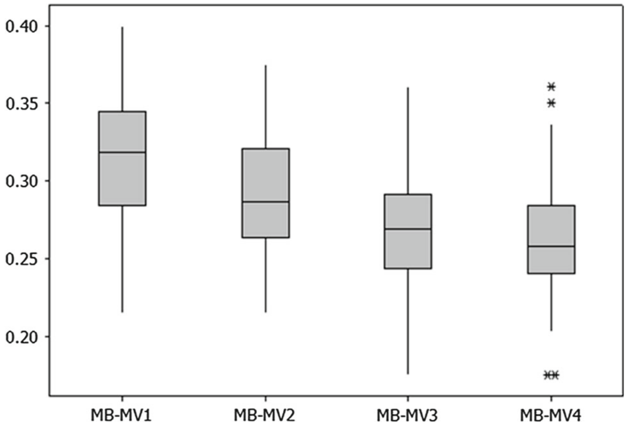


Fig. 27 *Boxplot* of *Euclidean* distances across the subsequent validation DoEs

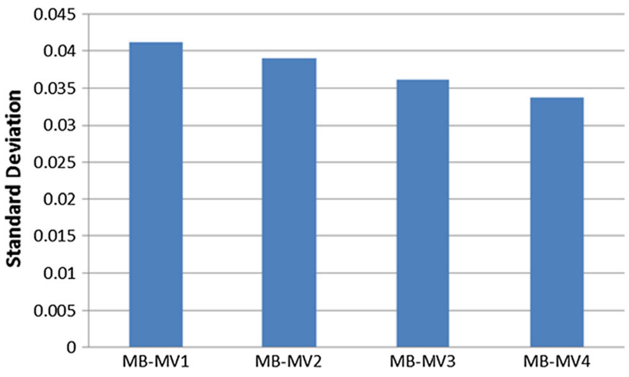


Fig. 28 Standard deviation (σ) of the Mm Distances

was negligible (i.e. $-0.05 \leq r \leq 0.05$), thus, the designs are quasi-orthogonal.

Figures 29, 30, 31 and 32 illustrate plots of 'PN' and 'Fuel Consumption' responses through stages MV1 and MV4. These figures clarify how the shape and trend of the responses, particularly the fuel consumption response, are transformed iteratively through collecting more infill test points, improving the prediction accuracy throughout the design space. As an example, it can be seen that the shape of fuel consumption response after collecting 4 sets of validation points (Fig. 32) is significantly different from the response model at sequence 1 (Fig. 31), especially at the extremes of the design space, i.e. the corner areas. Given that one of the main shortcomings of rigid OLH designs is to collect enough information at the areas around the boundary limits, using the sequential DoE method for this case study helped to collect more data around the unexplored areas next to the boundary limits, and consequently delivered a more accurate response model. Figures 33 and 34 illustrate the improvement in model accuracy through the MB–MV sequence in terms of the model prediction error (expressed as the ratio of validation RMSE to mean response, as percentage) for PN and fuel consumption. The decreasing trend in the validation RMSE shows that the quality of the response surfaces is enhanced, step by step. It can be seen that for the minimap point illustrated in Figs. 33 and 34, the relative validation RMSE is 1 % for 'Fuel Consumption' and 8 % for 'PN' after the 4th MB–MV iteration. Given the engineering target for model quality for fuel and PN responses of 1 and 10 %, respectively, it could be argued that for this case the engine response models were acceptable after the second MB–MV iteration, i.e. based on a mapping DoE of only 80 (65 MB + 15 MV) test points. This is significantly less than the normal mapping DoEs, which typically use 150 test points.

5 Summary, conclusions and future work

The aim of this paper was to present the development and validation of a permutation genetic algorithm for a sequential MB–MV DoE strategy based on OLHs. The motivation for the research was the complex engineering problem of a GDI engine mapping case study, for which an efficient DoE strategy is required to maximise the information gained with minimal resource expenditure, in terms of

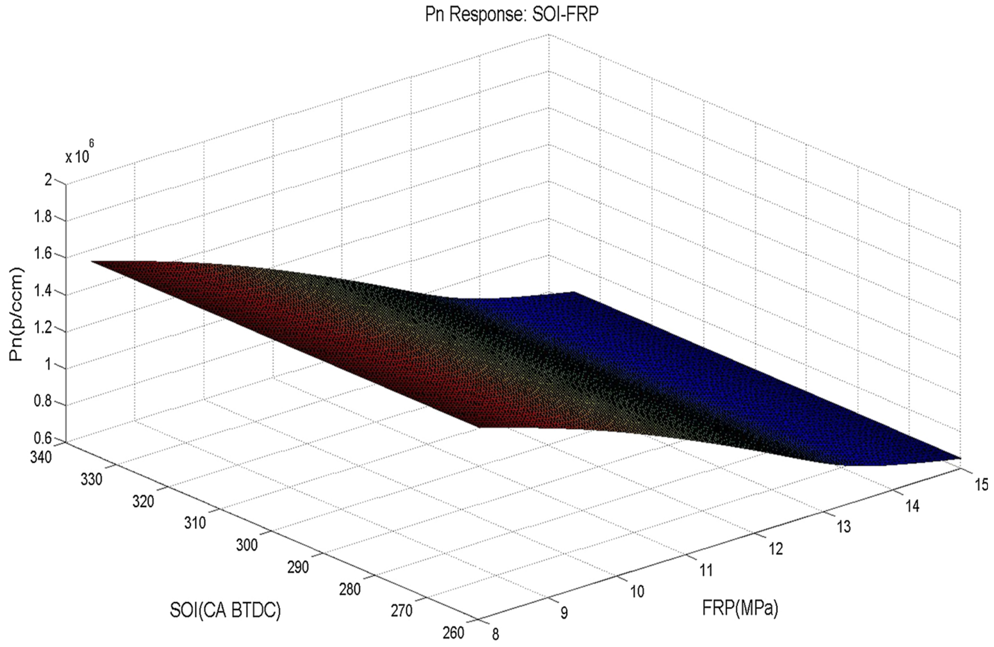


Fig. 29 PN response at MV1 Stage

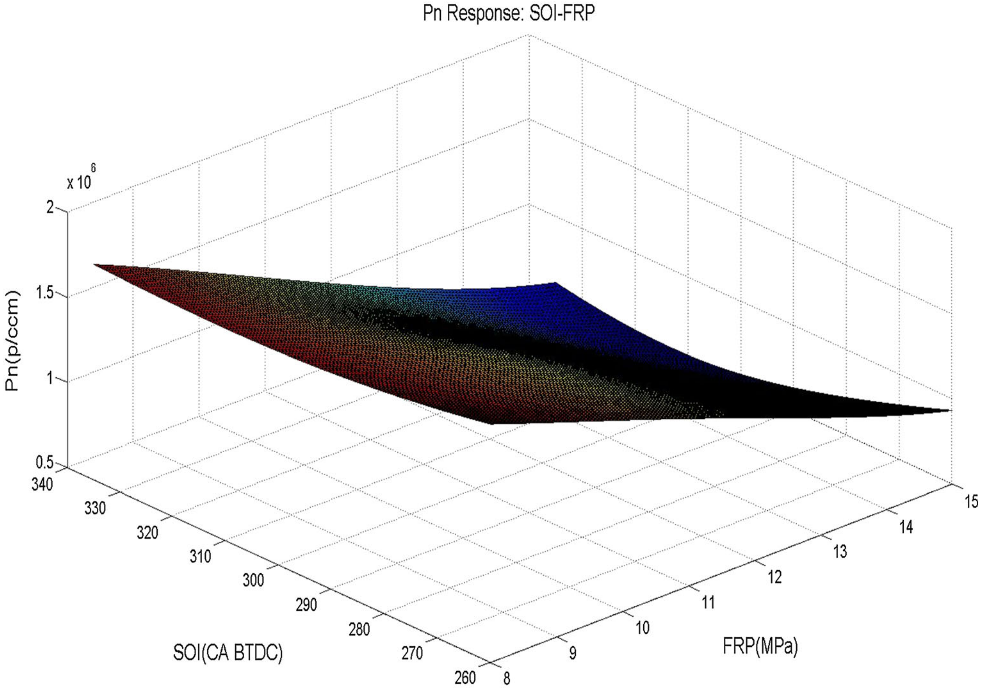


Fig. 30 PN response at MV4 Stage

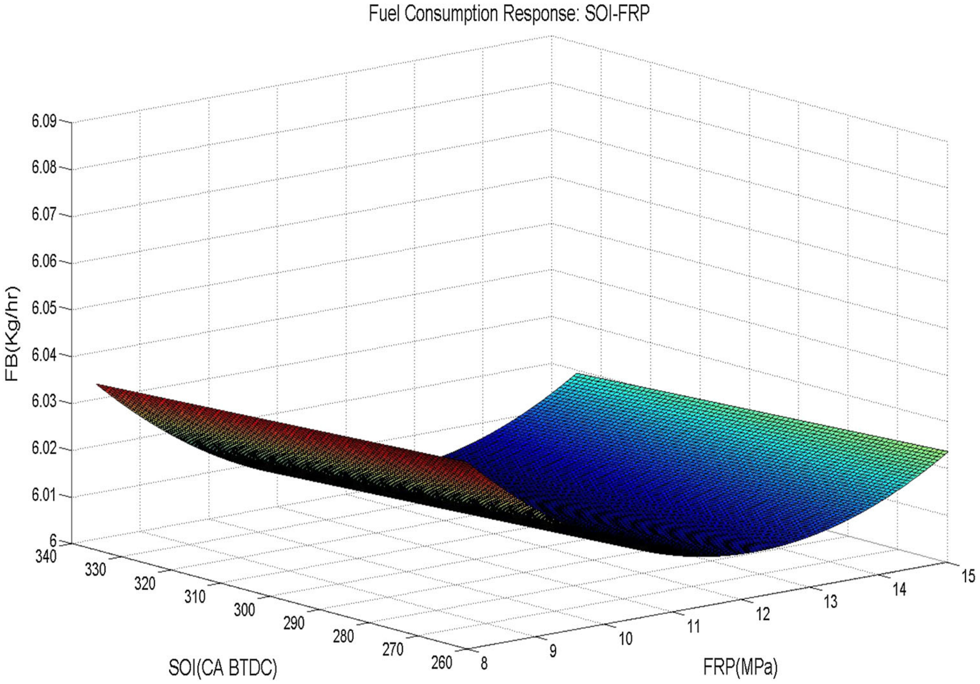


Fig. 31 Fuel consumption response at MV1 Stage

engine testing. Many other practical engineering problems, including those requiring computer-based experimentation, such as aerospace or automotive structural design based on finite elements simulation, where computation places significant challenges, could benefit from the application of the exploration-based sequential DoE methodology and the PermGA algorithm described in this paper.

The PermGA algorithm of Bates et al. (2003, 2004) was used as the basis for the development presented in this paper. The algorithm was modified principally by extensions

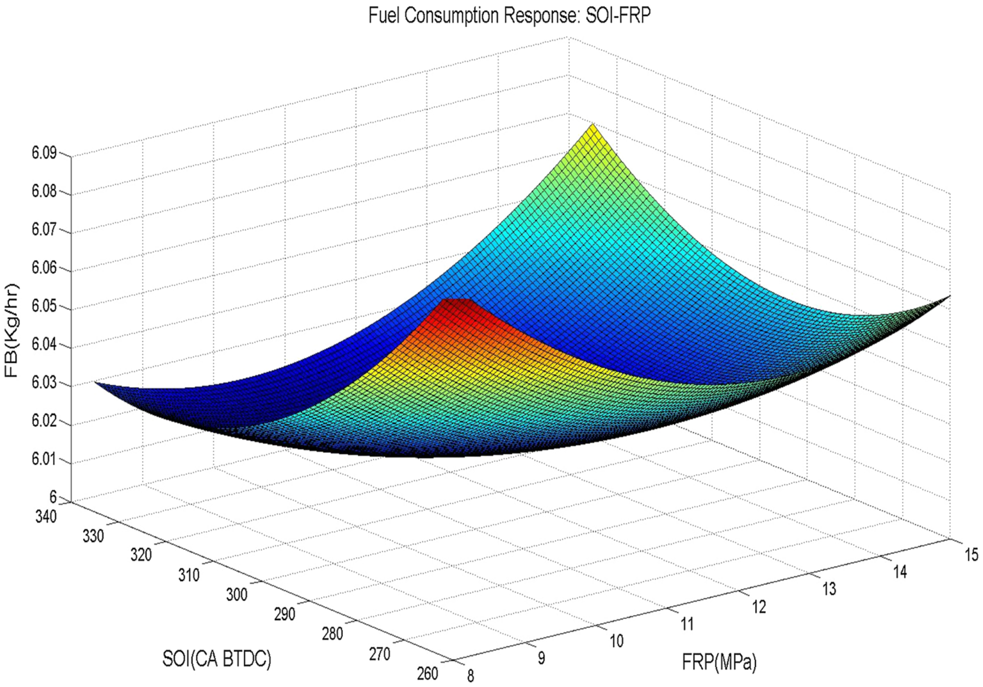


Fig. 32 Fuel consumption response at MV4 Stage

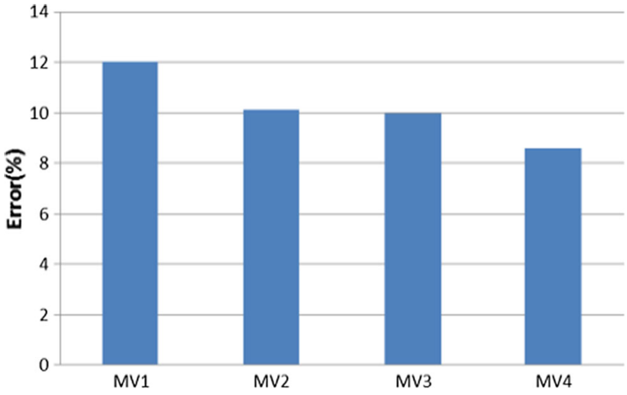


Fig. 33 PN model prediction error

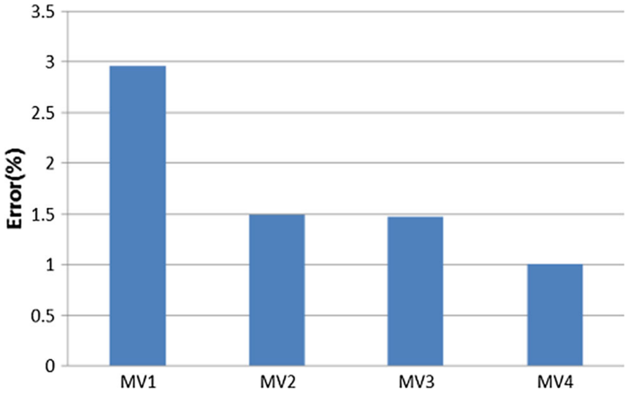


Fig. 34 Fuel consumption model prediction error

to the fitness function to enable the generation of flexible sequences of infill DoEs using the same principles of optimal Latin hypercubes, and preserving good space fillingness and statistical properties of the overall DoE. Significant further modification and extension of the PermGA algorithm was required to deal with non-orthogonal variables spaces. The introduction of the adaptive penalty function (SUMT) was proven to be effective in dealing with nonlinearly constrained design spaces to ensure that uniform space-filling DoE sequences can be achieved. The paper shows that this can work effectively even when the design space is progressively constrained (i.e. between MB–MV iterations). This is a very important feature as practical problems such as the engine mapping experiments reported by Dwyer et al. (2013), required revisions of the design space after the initial screening OLH DoE, based on both feasibility (e.g. combustion stability) and engineering preference (e.g. narrowing down of the variable space of interest to hone in on areas where optimal solutions appear more likely based on the analysis of trends).

The overall PermGA algorithm is presented in Fig. 2 as an XDSM graph, which adds clarity to the understanding of the flows compared to the conventional pseudocode or flow graph, and improves communication between the computation and design science communities.

Given that the motivation for this research is a real-world engineering problem, the validation of the methodology and algorithm developed was underpinned by the rigorous framework used in design science known as "the validation square" (Pedersen et al. 2000). Accordingly,

(i) The theoretical structural validity was based on a systematic analysis of the problem and the corresponding algorithm structural and logical requirements;
(ii) The empirical structural validity was demonstrated through Examples 1–4 presented in Sect. 3.2.2, which have demonstrated that the algorithm performs well under the range of test cases derived from the analysis of practical engineering problems;
(iii) The theoretical performance validity was pursued via a theoretical benchmark problem based on the SHCB function. The scenario considered a 4-step MB–MV sequence, with performance evaluated both in terms of uniformity of the overall DoE —which is directly related to the fitness function of the PermGA algorithm (hence validating the performance of the PermGA algorithm), and in terms of the improvement in the model quality measured in terms of PRESS RMSE—validating the MB–MV methodology based on PermGA;
(iv) The empirical performance validity was completed via the GDI engine mapping experiments case study, where engine dynamometer test data was collected based on a DoE plan generated using the PermGA algorithm presented in this paper. The case study results provided validation evidence for the PermGA-based MB–MV methodology. Furthermore, this case study has emphasised the complexities of real-world application of computational methodologies; e.g. validation of models must include phenomenological reasoning, and the observed behaviour along different response dimensions can have significantly different intrinsic characteristics (e.g. the combustion variability and measurement accuracy has a significant impact on the modelling fidelity that can be achieved for fuel flow compared to particulate numbers, as illustrated by the results in Figs. 33, 34).

As an overall conclusion, the work reported in this paper demonstrates that the developed PermGA and Infill PermGA algorithms can generate quasi-orthogonal uniformly distributed space-filing OLH DoEs, through sequential augmentation of a space-filling OLH DoE. The validation results have shown that a sequential MB–MV strategy is effective in generating models of required accuracy with a reduced testing sequence compared to the conventional approach used in practice, which is based on collecting one large DoE. By monitoring the modelling accuracy a model's accuracy within the MB–MV iteration, testing can be stopped if the models are sufficiently accurate, thus reducing unnecessary further testing which adds little additional information. Conversely, a further MV OLH DoE can be added if any of the response models is insufficiently accurate for prediction purposes.

Thus, it can be summarised that: (i) MB–MV reduces both testing and computational effort for achieving specified model accuracy (e.g. MB–MV reduced the total number of required test points for the GDI engine testing problem by up to 45 %, compared to the current practice), and (ii) provides robust scalable adaptivity to account for insufficient model accuracy. On this basis, it can be expected that this work will have significant impact in application to a broad range of engineering problems.

While the validation approach presented in this paper is complete from a design science framework viewpoint in the sense that theoretical/empirical and structural/performance aspects have been systematically considered, from a computational point of view further work is needed to formally address the efficiency of the PermGA algorithm.

Compliance with ethical standards

Conflict of interest The research work presented in this paper was funded by the UK Technology Strategy Board (TSB) through the Carbon Reduction through Engine Optimization (CREO) project.

References

Audze P, Eglais V (1977) New approach for planning out of experiments. Probl Dyn Strengths 35:104–107

Barbosa H, Lemonge AC, Bernardino HS (2015) A critical review of adaptive penalty techniques in evolutionary computation. In: Datta R, Deb K (eds) Evolutionary constrained optimization. Springer, Berlin, pp 1–27

Bates SJ, Sienz J, Langley DS (2003) Formulation of the Audze-Eglais Uniform Latin Hypercube design of experiments. Adv Eng Softw 34(8):493–506

Bates SJ, Sienz J, Toropov V (2004) Formulation of the optimal latin hypercube design of experiments using a permutation genetic algorithm. In: 45th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference, California

Bertram A (2014) A novel particle swarm and genetic algorithm hybrid method for improved heuristic optimization of diesel engine performance. Graduate Theses and Dissertations

Byrne CL (2012) Alternating minimization as sequential unconstrained minimization: a survey. J Optim Theory Appl 156(3):554–566

Coley DA (1999) An introduction to genetic algorithms for scientists and engineers. World Scientific, Singapore

Crombecq K, Dhaene T (2010) Generating sequential space-filling designs using genetic algorithms and Monte Carlo methods. In: Simulated evolution and learning. Springer, Berlin, pp 80–84

Crombecq K, Tommasi LD, Gorissen D (2009) A novel sequential design strategy for global surrogate modelling. In: Simulation conference, pp 731–742

Crombecq K, Laermans E, Dhaene T (2011) Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling. Eur J Oper Res 214(3):683–696

Deb K, Agrawal S (1999) Understanding interactions among genetic algorithm parameters. In: Foundations of genetic algorithms, pp 265–286

Deb M, Banerjee R, Majumder A, Sastry GR (2014) Multi objective optimization of performance parameters of a single cylinder diesel engine with hydrogen as a dual fuel using Pareto-based genetic algorithm. Int J Hydrog Energy 39(15):8063–8077

Dhingra S, Bhushan G, Dubey KK (2014) Multi-objective optimization of combustion, performance and emission parameters in a jatropha biodiesel engine using Non-dominated sorting genetic algorithm-II. Front Mech Eng 9(1):81–94

Draguljić D, Santner TJ, Dean AM (2012) Noncollapsing space-filling designs for bounded nonrectangular regions. Technometrics 54(2):169–178

Dwyer TP, Kianifar MR, Bradley WJ, Campean IF, Mason BA, Ebrahimi MK, Richardson D, Beddow L (2013) DoE framework for GDI engine mapping and calibration optimisation for CO2 and particulate number emissions. In: Roepke (ed) Design of experiments in engine development. Expert Verlag, Berlin, pp 418–432

Fang H, Mark FH (2005) Metamodeling with radial basis functions. In: 46th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference, Austin, Texas. AIAA article, pp 2005–2059

Forrester A, Sobester DA, Keane A (2008) Engineering design via surrogate modelling: a practical guide. Wiley, New York

Fuerle F, Sienz J (2011) Formulation of the Audze-Eglais uniform Latin hypercube design of experiments for constrained design spaces. Adv Eng Softw 42(9):680–689

Geest JD, Dhaene T, Fach N, Zutter DD (1999) Adaptive CAD-model building algorithm for general planar microwave structures. IEEE Trans Microw Theory Tech 47(9):1801–1809

Grefenstette J (1986) Optimization of control parameters for genetic algorithms. IEEE Trans Syst Man Cybern 16(1):122–128

Grove DM, Woods DC, Lewis SM (2004) Multifactor B-spline mixed models in designed experiments for the engine mapping problem. J Quality Technol 36(4):380–391

Hagen DF, Holiday G (1976) Effects of engine operating and design variables on exhaust emissions. SAE. Detroit, Michigan, pp 206–223

Hartmann B, Nelles O (2013) Adaptive test planning for the calibration of combustion engines—methodology. In: Design of experiments (DoE) in engine development, pp 1–16

Johnson ME, Moore LM, Ylvisaker D (1990) Minimax and maximin distance designs. J Stat Plan Inference 26(2):131–148

Joseph VR, Hung Y (2008) Orthogonal-maximin latin hypercube designs. Stat Sin 18(18):171–186

Khan MAZ (2011) Transient engine model for calibration using two-stage regression approach. PhD Thesis, Loughborough University

Kianifar MR, Campean IF, Richardson D (2013) Sequential DoE framework for steady state model based calibration. SAE Int J Engines 6(2):843–855

Kianifar MR, Campean F, Wood A (2014) PermGA algorithm for a sequential optimal space filling DoE framework. In: 2014 14th UK workshop on computational intelligence (UKCI). IEEE

Klein P, Kirschbaum F, Hartmann B, Bogachik Y, Nelles O (2013) Adaptive test planning for the calibration of combustion engines—application. In: Design of experiments (DoE) in engine development, pp 17–30

Kruse T, Kurz S, Lang T (2010) Modern statistical modeling and evolutionary optimization methods for the broad use in ECU calibration. In: Advances in automotive control, pp 739–743

Lam CQ (2008) Sequential adaptive designs in computer experiments for response surface model fit. PhD thesis Typescript, The Ohio State University

Lambe AB, Martins JRRA (2012) Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. Struct Multidiscip Optim 46(2):273–284

Liefvendahl M, Stocki R (2006) A study on algorithms for optimization of Latin hypercubes. J Stat Plan Inference 136(9):3231–3247

Liepins GE, Vose MD (1990) Representational issues in genetic optimization. J Exp Theor Artif Intell 2(2):101–115

McKay MD, Beckman RJ, Conover WJ (2000) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 42(1):55–61

Mezura-Montes E, Coello Coello CA (2011) Constraint-handling in nature-inspired numerical optimization: past, present and future. Swarm Evolut Comput 1(4):173–194

Michalewicz Z (1996) Genetic algorithms + data structures = evolution programs. Springer, Berlin

Michalewicz Z, Schoenauer M (1996) Evolutionary algorithms for constrained parameter optimization problems. Evolut Comput 4(1):1–32

Morris MD, Mitchell TJ (1995) Exploratory designs for computational experiments. J Stat Plan Inference 43(3):381–402

Morton TM, Knott S (2002) Radial basis functions for engine modelling. In: Statistics and analytical methods in automotive engineering. IMechE Paper C606/022/2002. London, UK

Narayanan A, Toropov VV, Wood AS, Campean IF (2007) Simultaneous model building and validation with uniform designs of experiments. Eng Optim 39(5):497–512

Owen AB (1992) Orthogonal arrays for computer experiments, integration and visualisation. Stat Sin 2:439–452

Pedersen K, Emblemsvag J, Bailey R, Allen JK, Mistree F (2000) Validating design methods and research: the validation square. In: ASME DETC 2000/DTM-14579, Proceedings of ASME design engineering technical conferences

Provost F, Jensen D, Oates T (1999) Efficient progressive sampling. In: Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining—KDD '99. ACM Press, New York, pp 23–32

Roepke K (2009) Design of experiments in engine development. Expert Verlag, Berlin

Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments. Stat Sci 4(4):409–423

Schwefel H (1993) Evolution and optimum seeking: the sixth generation. Wiley, New York

Seabrook J, Salamon T, Edwards S, Noell I (2003) A comparison of neural networks, stochastic process methods and radial basis function for the optimization of engine control parameters. In: Second conference design of experiments in engine development

Seabrook J, Collins J, Edwards S (2005) Application of advanced modelling techniques to the calibration of gasoline engines with direct injection and variable valve timing. In: Roepke K (ed) Design of experiments in engine development. Expert Verlag, Berlin, pp 235–245

Shukla PK, Deb K (2007) On finding multiple Pareto-optimal solutions using classical and evolutionary generating methods. Eur J Oper Res 181(3):1630–1652

Steinberg B, Lin D (2006) A construction method for orthogonal Latin hypercube designs. Biometrika 93(2):279–288

Tang B (1993) Orthogonal array-based latin hypercubes. J Am Stat Assoc 88(424):1392–1397

Van Dam E, Husslage B, Hertog D, Melissen H (2007) Maximin latin hypercube designs in two dimensions. Oper Res 55(1):158–169

Wang L, Shan S, Wang GG (2004) Mode-pursuing sampling method for global optimization on expensive black-box functions. Eng Optim 36(4):419–438

Ye KQ, Li W, Sudjianto A (2000) Algorithmic construction of optimal symmetric Latin hypercube designs. J Stat Plan Inference 90(1):145–159