A METHOD TO IMPROVE DESIGN RELIABILITY USING OPTIMAL LATIN HYPERCUBE SAMPLING

Rafał Stocki

Institute of Fundamental Technological Research,
Polish Academy of Sciences,
Swietokrzyska 21, 00-049 Warsaw, Poland

Abstract

In the paper an algorithm for design reliability improvement is proposed. Its key part consists in the computation of the correlations between constraint functions and design variables which are subsequently used to find the new design iteration. It is shown that the optimal Latin hypercube (OLH) sampling provides an extremely efficient technique for assessing the values of correlation coefficients. Since finding the large OLH designs is not a trivial task, a study on the OLH generation algorithms was performed. Two algorithms were found to be particularly effective, namely, the columnwise-pairwise algorithm and the genetic algorithm.

The presented strategy proves to be especially useful when alternative gradient-based methods cannot be used, which is often the case for computationally expensive problems involving noisy and highly nonlinear responses. The method is best suited for problems where the probability of failure for the initial design is large and the main interest is to find a more reliable design rather than the optimal one in the sense of reliability-based optimization.

The method is illustrated with two numerical examples. One model example and one concerning the problem of thin-walled beam crash.

Keywords: optimal Latin hypercube sampling, reliability-based optimization, genetic algorithms, crashworthiness reliability

1 Introduction

Due to the continuous development of computational technology we are now able to build and analyze more and more refined models of complex nonlinear physical phenomena, e.g. simulate the crash of a car. Finite element models can now include hundreds of thousands or even millions of elements and powerful parallel machines are used for simulations. However, what is becoming commonly shared opinion, by not investigating the influence of unavoidable randomness of model parameters we loose important information concerning the underlying problem. This may lead in effect to potentially failure prone design. Designs that perform well for nominal values of their parameters sometimes turn out to be completely unreliable when the imperfections of the model parameters and operating conditions are taken into account.

Such a behavior manifests itself particularly for designs resulting from deterministic optimization and especially in the optimization for crashworthiness. However, due to limitations of technology and computer resources researchers do not address these problems very often. The papers dedicated to optimizations in crashworthiness related problems that were published in the last years, deal almost exclusively with deterministic, response surface based optimization, see [2, 23, 18, 4, 21].

A remedy for the high probability of failure of the deterministic optimal design can be reliability-based optimization (RBO). The RBO problem may be formulated in several ways [10, 8]. One way is to minimize the initial structural cost under the constraints imposed on the values of probabilities of failure corresponding to various limit states, another way is to maximize the system reliability under design constraints. RBO algorithms that usually employ first order reliability method (FORM) concepts, see [13], proved to be efficient in many structural optimization problems but their direct application to crashworthiness optimization problems seems to be rather unrealistic. First attempts of RBO for crashworthiness were almost always based on some response-surface approximation of the structural response (see eg. [6]).

The method that is proposed in the current paper is aimed at fast improvement of the design reliability using a random sampling technique and is especially efficient when only a moderate number of sample points can be afforded. Though formulated as a method to solve the RBO problem, the solution technique and the scope of potential applications make the term "design reliability improvement method" more appropriate than "RBO method".

The response surface method (RSM) which, when cleverly used, is an excellent tool in solving many optimization problems may be less reliable for extremely noisy designs, or if there is a high degree of nonlinearity, especially instability. Since the crash analysis was meant as the main application of the method presented in this paper, however being far from very strong opinions that question the very idea of RSM-based crashworthiness optimization [11], it was deliberately decided not to use the RSM concept. In the paper [23], Stander et al. compared various RSM and random search techniques for the deterministic optimization. The random search was based on Latin Hypercube (LH) sampling. It was shown that for some examples, even using small samples the obtained results may be surprisingly good when compared to RSM-based optimization.

The sampling technique which is the key part of the proposed method is the so-called optimal Latin hypercube (OLH) sampling. As it will be described in detail in section 3, the rearrangement of LH points which minimizes an appropriate criterion can produce the design of experiments that is particularly efficient in predicting statistical properties of a model response. It was proposed to use linear correlation coefficients as a measure of dependency between model responses (optimization constraints) and design variables. They are then used to determine the change direction and to select the new design iteration. The OLH sampling allows to assess the values of correlation coefficient with acceptable accuracy even for small samples. In the sections 3.1 and 3.2 are presented two efficient algorithms to create large and medium size OLHs.

Up to now there have been few attempts to use LHs and OLHs in reliability analysis. For example, in [16] Olsson et al. used LH in the framework of importance sampling method to improve the efficiency of FORM results, and in [25] optimal symmetric Latin hypercube was used as the plan of experiments to generate response surfaces that were subsequently used in the most-probable-point search algorithm. However, both these techniques are much too expensive for our purposes.

The two examples (sections 4 and 5), which illustrate the method, were solved with the M-Xplore module of the Radioss software [3], co-developed by the author.

2 Presentation of the method

2.1 Problem formulation

Let $\mathbf{X} = \{X_1, X_2, \ldots, X_N\}$ be the vector of independent random variables representing uncertainties of selected system parameters like: material parameters, geometry, loads, initial and boundary conditions. Each random variable X_i, $i = 1, \ldots, N$, is described by its probability density function (PDF) with the corresponding mean value μ_{X_i} and standard deviation σ_{X_i}. Let us then denote by μ the vector of mean values of the random variables

$$\mu = \{\mu_{X_1}, \mu_{X_2}, \dots, \mu_{X_N}\}. \tag{1}$$

Assume now that n out of N mean values are the design variables in our design optimization (design improvement) problem. To facilitate the formulation it is assumed that the random variables are ordered in such a way that the first n elements in μ correspond to the design variables $\mathbf{y}$. It is then

$$\mathbf{y} = \{y_1 = \mu_{X_1}, y_2 = \mu_{X_2}, \dots, y_n = \mu_{X_n}\}, \quad n \leq N. \tag{2}$$

Depending on the nature of a random variable X_i, $i = 1, \dots, n$, its standard deviation may be kept fixed during the optimization/improvement process or changed according to a fixed coefficient of variation $\nu_{X_i} = \sigma_{X_i}/\mu_{X_i}$.
The optimization problem can now be stated as follows:

$$\begin{aligned}
&\text{find} && \mathbf{y} \in \mathbb{R}^n, && && (3)\\
&\text{that minimizes} && f(\mathbf{y}), && && (4)\\
&\text{subject to:} && \mathbb{P}[c_i(\mathbf{X}) \geq 0] \geq p_i^a, && i = 1, \dots, m, && (5)\\
& && {}^l y_j \leq y_j \leq {}^u y_j, && j = 1, \dots, n, && (6)
\end{aligned}$$

where $f(\mathbf{y})$ is the objective function, $c_i(\mathbf{x})$ are the constraint functions, $\mathbb{P}(\,\cdot\,)$ is the probability function, p_i^a are the admissible probabilities and ${}^l y_j$, ${}^u y_j$ are the lower and upper bounds, respectively, imposed on the design variables.
The equations (5) set up the so-called reliability constraints. They state that the constraints $c_i(\mathbf{x}) \geq 0$, $i = 1, \dots, m$, have to be fulfilled with a certain probability p_i^a or that the respective probabilities of failure, P_{f_i}, have to be smaller than $1 - p_i^a$. However, it must be stressed that using a Monte Carlo type simulation, when only small samples can be afforded and for problems where high reliabilities are required, will result in very poor approximations of the values of reliability constraints. For example, deciding to use less then 100 sample points (which will usually be the case in 'real life' applications) and setting $p^a = 0.99$ the constraints (5) should rather be read as: $c_i(\mathbf{x}) \geq 0$ for all the points (lowercase $\mathbf{x}$ stands for samples of $\mathbf{X}$). In fact, to guarantee a good estimation (say, coefficient of variation of the estimator smaller than 0.1) of the probability of failure using crude Monte Carlo sampling, for the expected P_f value of about 0.01, 10000 sample points should be generated. As it will be shown later in the text the OLH sampling allows to reduce the sample size but still, the accurate approximation of the reliability constraints is not possible if we sample from the original distributions of random variables (contrary to importance sampling technique, see [16]). However, since our aim in this paper is rather to find a more reliable design than to find the optimal one we decided to accept the limited accuracy of the adopted approach.

2.2 Solution algorithm

The solution method can be classified as a probabilistic search algorithm. To find the starting point and subsequent points in the iteration process the sampling technique, based on the OLH design is employed, (see section 3 for details). The principal idea of the method is based on the assumption that the next design point is chosen from among the points for which the values of objective function and the constraints $c_i(\mathbf{x})$, $i = 1, \dots, m$, were already computed. By doing this, as opposed to the methods based on the concept of response surface, we deliberately avoid any assumptions concerning the type of function approximating the objective and constraints. This approach is similar to the one proposed by Marczyk [11] and implemented in STORM [22] software.

2.2.1 Choice of the starting point

Depending on having some prior knowledge of the problem, the starting point can be either directly specified or it can be chosen by examining the results for points generated using OLH sampling. In the latter case the PDF's of random variables corresponding to the design variables are temporarily changed to uniform PDF's with the bounds defined by ${}^{l}y_i$, ${}^{u}y_i$, $i = 1, \ldots, n$. Next, a sample of $\mathbf{X}$ is generated and the feasible sample point that minimizes the objective function is chosen as the starting point. This means that the mean values of variables X_i, $i = 1, \ldots, n$, are shifted to the starting point. Next, the modified PDF's are changed back to the original ones with the new mean values and the original standard deviations. In the case of constant coefficients of variation the standard deviations should also be recomputed.

A reason for the initial PDF's modification is to be able to cover the entire design space with the samples in order to find a good starting point. As will be described later in the text (section 3.3) the special, modified OLH design is used at this stage.

2.2.2 Improvement strategy

At each iteration step of the algorithm K sample points are generated using the OLH design. Now, the problem to be solved is to choose from among these points the next design. The selection strategy can be summarized as follows:

First, the values of reliability constraints (5) are estimated, and those which are violated are identified. Let I be the set of indices corresponding to these constraints

$$I = \{i : 1 \leq i \leq m,\ \mathbb{P}[c_i(\mathbf{X}) \geq 0] < p_i^a\}. \tag{7}$$

Next, taking into account only the last sample's results the correlation coefficients between the random variables X_i, $i = 1, \ldots, n$, and the constraint functions $c_j(\mathbf{x})$, $j \in I$, are computed. They are given by

$$\rho_{ij} = \frac{\sum_{k=1}^{K} [x_i^{(k)} - \bar{x}_i][c_j(\mathbf{x}^{(k)}) - \bar{c}_j]}{\sqrt{\sum_{k=1}^{K} [x_i^{(k)} - \bar{x}_i]^2} \sqrt{\sum_{k=1}^{K} [c_j(\mathbf{x}^{(k)}) - \bar{c}_j]^2}}, \qquad i = 1, \ldots, n,\ \ j \in I, \tag{8}$$

where $\rho_{ij} = \rho(X_i, c_j)$, $\bar{x}_i$ and $\bar{c}_j$ are the sample means and $x_i^{(k)}$ denotes the i-th component of the k-th realization (k-th sample point) of the random vector $\mathbf{X}$. The computed correlation coefficients play the key role in selecting the change 'direction' or, more precisely, selecting the subset of sample points from which the new design will be chosen. Some points $\mathbf{x}^{(k)}$, $k = 1, \ldots, K$, can be immediately eliminated if they violate simple bounds (6) or if $c_i(\mathbf{x}^{(k)}) < 0$, $i = 1, \ldots, m$. To facilitate the presentation let us denote by $\tilde{I}$ the set of indices corresponding to the points fulfilling these criteria, that is

$$\tilde{I} = \{k :\ 1 \leq k \leq K,\ \ c_i(\mathbf{x}^{(k)}) \geq 0,\ \ i = 1, \ldots, m \ \text{ and }\ {}^{l}y_j \leq x_j^{(k)} \leq {}^{u}y_j,\ \ j = 1, \ldots, n\}. \tag{9}$$

Now, from the set of points $\mathbf{x}^{(i)}$, $i \in \tilde{I}$ we want to choose the points which are the most likely to 'improve' the reliability constraints. In order to do this for each design variable the following expression is computed

$$d_i = \sum_{j \in I} \rho(X_i, c_j) \Big(1 + \mathbb{P}[c_j(\mathbf{X}) < 0] \Big), \quad i = 1, \ldots, n. \tag{10}$$

If d_i is positive then only the sample points which have the value of the i-th variable greater then the corresponding mean value μ_{X_i} are considered, and similarly if $d_i < 0$ only points for which $x_i^{(k)} < \mu_{X_i}$ are taken into account. If $d_i = 0$ then all the points $\mathbf{x}^{(i)}$, $i \in \tilde{I}$, can be considered. The idea behind the expression (10) is based on the information carried by correlation coefficients. With the values changing from -1 to 1 they provide a measure of linear dependence between the random variables and the constraint functions. The weight factor in parenthesis accounts for the extent of violation of a particular reliability constraint.

Due to the nonlinear character of the constraint functions it can happen that the described procedure will exclude some points that might also be considered as potential 'candidates' for the next design iteration. To avoid this, in addition to the points already chosen from the sample, we select (if they exist) those for which the values of constraints c_i, $i \in I$, are greater than the corresponding ones for the point selected in the previous step as the current design.

To illustrate the above procedure let us consider an example with 3 design variables where at a typical iteration step 50 sample points are generated. The values of admissible probabilities p^a(cf. (5)) are the same for all the constraints and equal 0.99. Assume next that four constraints, $c_i(\mathbf{x})$, $i = 1, \ldots, 4$, are violated by 20, 10, 3 and 15 samples, respectively. Computing first the correlation coefficients (values in the table below), the corresponding d_i, $i = 1, \ldots, 3$, values are subsequently computed.

	c_1	c_2	c_3	c_4
X_1	-0.6	-0.1	0.8	-0.2
X_2	0.01	0.5	-0.9	-0.3
X_3	0.65	0.2	-0.3	0.1

$$\begin{aligned} d_1 &= -0.6\Big(1+\frac{20}{50}\Big) - 0.1\Big(1+\frac{10}{50}\Big) + 0.8\Big(1+\frac{3}{50}\Big) - 0.2\Big(1+\frac{15}{50}\Big) = -0.37, \\ d_2 &= 0.01\Big(1+\frac{20}{50}\Big) + 0.5\Big(1+\frac{10}{50}\Big) - 0.9\Big(1+\frac{3}{50}\Big) - 0.3\Big(1+\frac{15}{50}\Big) = -0.73, \\ d_3 &= 0.65\Big(1+\frac{20}{50}\Big) + 0.2\Big(1+\frac{10}{50}\Big) - 0.3\Big(1+\frac{3}{50}\Big) + 0.1\Big(1+\frac{15}{50}\Big) = 0.96. \end{aligned}$$

According to the adopted strategy one will choose the next design point from among points $\mathbf{x}^{(i)}$, $i \in \tilde{I}$, for which $x_1^{(i)} < \mu_{X_1}$, $x_2^{(i)} < \mu_{X_2}$ and $x_3^{(i)} > \mu_{X_3}$ and from the additional points selected as it was described above.

For a big number of design variables it is likely to happen that there will be no sample points satisfying all the criteria. In such a case additional points (at least one) should be generated in the region defined by the d_i values. Of course, a newly generated point can be accepted as the new design if it satisfies all the constraints (see (9)).

After determining the set of candidate sample points a criterion must be employed to choose the 'best' one. In the examples presented later in the text the following function was used as a criterion

$$h(\mathbf{x}) = w_f \tilde{f}(\mathbf{x}) - \sum_{i=1}^{m} w_i \tilde{c}_i(\mathbf{x}), \tag{11}$$

where $\tilde{f}(\mathbf{x})$ and $\tilde{c}_i(\mathbf{x})$, $i = 1, \ldots, m$, are the normalized values of the objective function and constraints, respectively, and w_f and w_i are the weight factors. The sample point $\hat{\mathbf{x}}$ that minimizes the function h is taken as the next design, or more precisely, $y_i = \hat{x}_i$, $i = 1, \ldots, n$.

There is some freedom in selecting the weight factors. For example, defining w_f as

$$w_f = \exp\left(1 - \frac{\mathbb{P}[c_k(\mathbf{X}) < 0]}{1 - p_k^a}\right), \tag{12}$$

where k is the index of the most violated reliability constraint, preferences points more likely to fulfill reliability constraints. The same is achieved by the following definition of the w_i factors:

$$w_i = 1 + \mathbb{P}[c_i(\mathbf{X}) < 0]. \tag{13}$$

Neglecting 1 in the above formula leads in practice to taking into account only violated reliability constraints. The choice of weight factors may also be determined by the computational cost of the underlying analysis. If one cannot afford to many sample points and its main aim is to make the initial design more reliable then w_f factor should be set to zero and the process stopped after finding the first design satisfying reliability constraints.

3 Efficient sampling techniques

A good samples generation technique is a crucial component of the presented optimization algorithm. It is especially important for computationally expensive problems. In order to avoid clustering of the sample points and to assure good estimation of the statistical moments of response functions the OLH sampling technique has been selected. Later in this section the two methods for building large OLH are presented, the columnwise-pairwise (CP) algorithm invented by Park [17] (with the modification described in [24]) and the genetic algorithm, inspired by the algorithm proposed in [20]. They were implemented in the M-Xplore module of the Radioss software [3] and explained in details in [9]. Below, only the most important information on the OLH design and the algorithms is given.

OLH is a viable sampling technique when one considers statistical optimality and projection properties. There are many criteria of the statistical optimality of a design of experiments. Most of them are based on fitting a (stochastic) model to experiments/computed data, see eg. [14] and [17]. Another criterion, which is of interest in the current paper, measures how well the statistical properties of some model are predicted. By good projection properties we mean here that the sample points are well spread out when projected onto a subspace spanned by a number of coordinate axes. This is very often desired in the applications when one does not know a priori if some random variables have a negligible effect on the response of the system.

A Latin hypercube is represented by $K \times N$-matrix (i.e. a matrix with K rows and N columns) $\mathbf{L}$ where each column of $\mathbf{L}$ consists of a permutation of the integers 1 to K. We will refer to each row of $\mathbf{L}$ as a (discrete) sample point and use the notation

$$\mathbf{L} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_K \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & & \vdots \\ x_{K1} & \cdots & x_{KN} \end{bmatrix}, \tag{14}$$

where $\mathbf{x}_i$ is the i-th sample point.

The LH design obtained by simply generating N random permutations of the numbers 1 to K and placing them as columns in the matrix, without any subsequent changes, will be referred later in the text as random Latin hypercube (RLH).

The matrix $\mathbf{L}$ can now be used to generate 'real' samples of the random vector $\mathbf{X}$ taking into account the distribution of each variable. To find the realization $(\mathbf{x}_i)_j$ of the random variable X_j, $1 \leq j \leq N$, corresponding to the number i, $1 \leq i \leq K$, in the j-th column of the matrix $\mathbf{L}$, the cumulative distribution function (CDF) of X_j is used

$$(\mathbf{x}_i)_j = F_{X_j}^{-1}(\tilde{x}_i), \tag{15}$$

where

$$\tilde{x}_i = \frac{i}{K} - \frac{1}{2K}. \tag{16}$$

In other words, the range of variability of each random variable is divided into K intervals of equal probability and the values $(\mathbf{x}_i)_j$, $i = 1, \ldots, K$ correspond to the medians of X_j in these intervals. Another choice, instead of (15), is to choose the $(\mathbf{x}_i)_j$ randomly in the i-th interval.

It is important to mention that in general the random variables can be arbitrarily distributed and correlated. However, to use the sample design generated with LH the variables must be first numerically transformed to a set of uncorrelated random variables. In the case when the joint probability density function is known the Rosenblatt transformation [19] can be used and when only marginal CDFs of the variables and the correlation matrix are known one may employ the Nataf transformation [15]. Both transform the original variables to the space of independent standardized Gaussian variables. The values of the random variables found in the transformed space using (15) are next transformed back to the original random variables $\mathbf{X}$.

The criterion that is used in the current paper to optimize LH design was proposed by Audze and Eglais in [1]. It is based on the function G which, in a physical analogy, is the sum of the norms of the repulsive forces if the samples are considered as electrically charged particles

$$G(\mathbf{L}) := \sum_{i=1}^{K} \sum_{j=i+1}^{K} \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|^2}. \tag{17}$$

From the point of view of the physical analogy, it would have been natural with the power 1 (instead of 2) in the denominator of the terms of the sum. However, with the power 2 a computation of a square root for each term is avoided. This has a noticeable effect on the execution speed since the function G will be evaluated many times in the inner loop during optimization. Using the function G the criterion that will allow to compare two LH designs can be stated as

$$\mathbf{L}_1 \text{ is better than } \mathbf{L}_2 \text{ if } G(\mathbf{L}_1) < G(\mathbf{L}_2). \tag{18}$$

As it was shown in [9] this criterion is a reasonable compromise between good statistical properties and efficiency.

Although for small Latin hypercubes the computational cost of finding OLH is negligible compared to the expensive computer simulation of a physical phenomenon, it grows very fast with the sample size and the number of variables. For large LH (hundreds of sample points and tens of variables) it may even take hours and days with fast computers. The computational cost depends of course on the algorithms used for OLH optimization and the adopted optimality criterion. In the next two sections the main features of the CP algorithm and the genetic algorithm are presented.

3.1 Columnwise-pairwise (CP) algorithm

The CP algorithm can be described in pseudo-code as follows:

```
Generate a random Latin hypercube:  $L_{new}$
$stop$ = FALSE
**while**( not $stop$ )
   $L_{old}$ = $L_{new}$
   Do a CP-sweep to generate $L_{new}$ from $L_{old}$
   **if** stopping criterion fulfilled
   **then** $stop$ = TRUE
**end of while-loop**
```

The so-called ‘CP-sweep’ operation is given as follows:

```
**for** $i$ = 1 **to** $N$
   Find the best first order modification of column $i$ and replace column $i$ with it
**end of for-loop**
```

where the first order modification of a column of a LH matrix is defined as an interchange of two of its elements. Of course, ‘best’ first order modification means the modification which gives the best LH according to the chosen criterion. The name CP-sweep indicates that it is a systematic procedure going through (sweeping) all columns of the LH-matrix testing interchanges.
The stopping criterion in the k-th step is given by the inequality

$$\Delta G_k < \epsilon \Delta G_1, \tag{19}$$

where ΔG_1 is the improvement in the first step and ϵ is a chosen parameter.
It can be shown [9] that the computational complexity of the CP algorithm is estimated by the expression

$$T_{\text{tot}} \sim NK^3(c_1 KN + c_2 K^2), \tag{20}$$

where T_{tot} stands for the total complexity of the algorithm and $c1$ and $c2$ are constants. It must be emphasized that the execution time is very sensitive to variations in K. A number of tests have been performed to estimate the speed of the computations. These include the cases $K = 50, 60, \ldots, 100$ for $N = 3$ which indicate an execution time $T \sim K^q$ with q between 4.5 and 5.5. To illustrate the order of magnitude of the growth one can extrapolate from the execution time of approximately 3 minutes for the 100×3 OLH on an SGI Octane2 workstation, assuming $T \sim N^5$. This gives an execution time for 1000×3 OLH of eight months.

3.2 Genetic algorithm

The genetic algorithm is a very general optimization technique and can be applied to a large class of optimization problems. In the general form it can be described in pseudo-code as follows

```
Generate initial population
Calculate fitness for individuals in the initial population
$stop$ = FALSE
**while**( not $stop$ )
```

```
        Select 'survivors'
        Cross-over the 'survivors'
        Mutate the resulting population
        Calculate the fitness of the new population
        **if** stopping criterion fulfilled
        **then** $stop$ = TRUE
    **end of while-loop**
```

The step of producing one new generation in the genetic algorithm is cheap compared to the CP-sweep. Also there is the possibility that the best LH in the new generation is not an improvement even if further iterations gives important improvements. For these reasons in the stopping criterion the accumulated improvement in the first s (say $s = 50$ or 100) generations is saved (not only the first improvement like in (19))

$$\Delta\tilde{G}_s = G(\mathbf{L}_s) - G(\mathbf{L}_0), \tag{21}$$

where $\mathbf{L}_s$ is the best LH in the s-th generation. Then in generation k, if k is a multiple of s the following inequality is checked

$$G(\mathbf{L}_k) - G(\mathbf{L}_{k-s}) < \epsilon\Delta\tilde{G}_s. \tag{22}$$

There are many variations of the genetic algorithm according to how one chooses to define the steps of selection, cross-over and mutation, also the initial population can be chosen in various ways. We now turn to the description of these steps for the problem of optimizing Latin hypercubes.

Initial population. We start by generating N_{pop} of random LHs which constitute the initial population. The number N_{pop} is required to be even because of the selection step.

Selection. The $N_{\text{pop}}/2$ best LH are chosen as 'survivors' and the rest is thrown away.

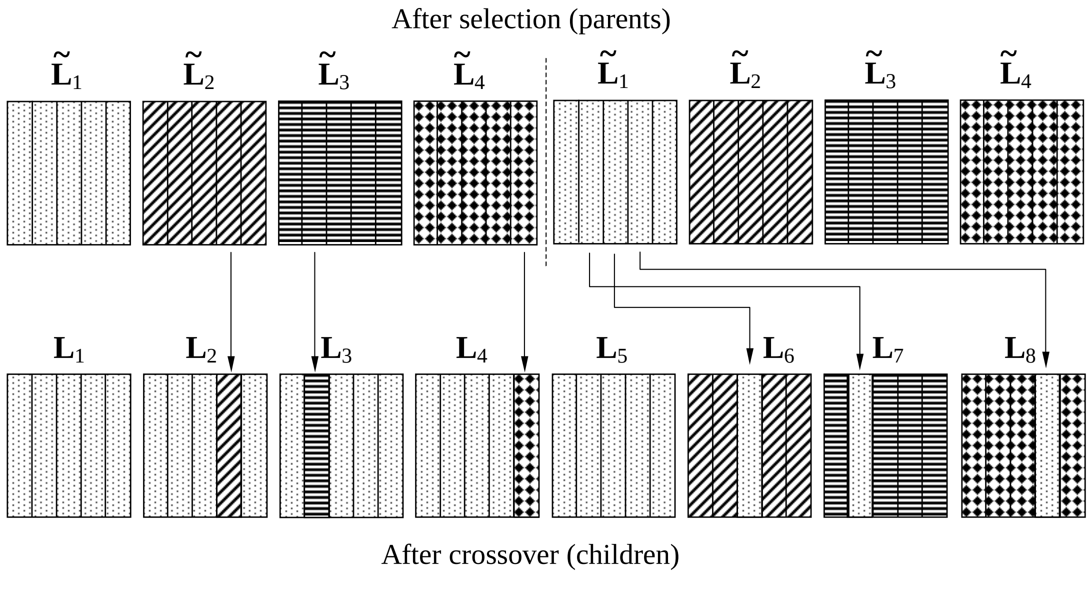

Figure 1: Illustration of the cross-over step. Observe that after cross-over $\mathbf{L}_1$ and $\mathbf{L}_5$ will be identical to $\tilde{\mathbf{L}}_1$ before cross-over, the best LH. Later, in the mutation step, $\mathbf{L}_5$ is allowed to change but not $\mathbf{L}_1$, so at least one copy of the best LH is always kept.

Cross-over. In this step N_{pop} children of the $N_{\text{pop}}/2$ survivors are generated. This step is illustrated in Fig. 1. First, the best LH is put as number 1 and $N_{\text{pop}}/2 + 1$ among the total set of children. Then for $k \in [2, N_{\text{pop}}/2]$ the best LH is mated with the k-th in the following way. The best and the k-th will generate two children. The first child is obtained by taking the best LH and replacing a random column with the corresponding column in the k-th LH. The resulting child is given the number k among the children. The second child is obtained by taking the k-th LH and replacing a random column with the corresponding column of the best LH. The resulting child is given number $N_{\text{pop}}/2 + k$ among the children.

Mutation. Mutation is done on all except the best LH from the earlier generation in the following way: for each column a random number in $[0, 1]$ is generated (according to the uniform distribution), if this is lower than a threshold p_{mut}, then two randomly chosen elements in the column are swapped.

As it was shown in [9] depending on the size of LH the CP algorithm or the genetic algorithm are found to be most competitive. Numerical experiments suggest the use of CP for K less than 150 and the genetic algorithm for larger problems. For the genetic algorithm the optimal values for the population size and the probability of mutation depend, of course, on the size of the problem. However, as a robust choice, $N_{\text{pop}} = 50$ and $p_{\text{mut}} = 0.1$ can be recommended.

3.3 Modified OLH design

Latin hypercube ensures good stratification of the design space when projected into 1-dimension. In addition, the OLH design avoids clustering of the sample points. However, it is not possible to eliminate clustering when projecting sample points of the N-dimensional OLH into less dimensions. In Fig. 2 two Latin hypercubes are presented. The left one is the OLH of 50 points in 2 dimensions. It can be seen that the points are uniformly scattered in the square. The right LH is the projection of the 50 points 4 dimensional OLH into 2D. In this case the clustering of sample points is clearly visible.

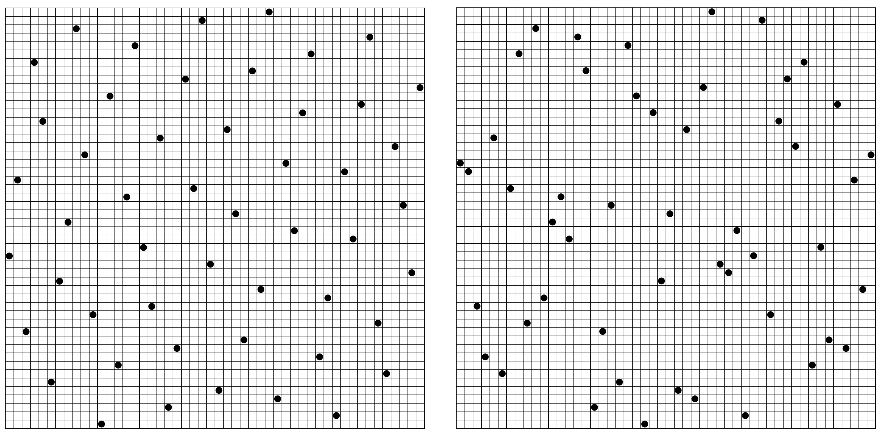

Figure 2: Left: 50×2 OLH. Right: Projection of the 50×4 OLH into 2 dimensions.

This feature of the OLH design is particularly unwanted when choosing the starting point for our algorithm (see section 2.2.1). Since the final design depends very much on the appropriate choice of the starting point, it is important to ‘cover’ the space of the design variable as good as possible. On the other hand the idea of the presented optimization/improvement approach is to account all the time for the uncertainties of the other parameters (not corresponding to the design variables). In order to find a compromise, the modified optimal Latin hypercube design (referred as MOLH) was proposed. It is optimal in the subspace of design variables, and the coordinates of sample points corresponding to remaining $N - n$ random variables are selected using either the CP or the genetic algorithm with the additional constraint that the optimality of the LH projection on the design space must be preserved.

From the implementation point of view this modification is quite straightforward. First, the 'design' OLH is generated, i.e. $K \times n$ matrix $\mathbf{L}_d$ is found. Next, in the $K \times N$ matrix $\mathbf{L}$, being the starting RLH for CP algorithm (or in initial population of RLHs for genetic algorithm), the columns corresponding to design variables are replaced by those from $\mathbf{L}_d$. Then, the algorithms described in the previous sections are employed with the following modifications: in the CP-sweep the columns 'pasted' from $\mathbf{L}_d$ are not considered, while in the genetic algorithm the cross-over and mutation operations do not affect columns from $\mathbf{L}_d$, their position in the matrix $\mathbf{L}$ and their contents remain unchanged.

3.4 Example: Efficiency of the OLH sampling

In order to compare the efficiency of different sampling techniques for the estimation of statistical properties let us consider the so-called Rosenbrock function b defined as

$$b(X_1, X_2) = 100(X_2 - X_1^2)^2 + (1 - X_1)^2. \tag{23}$$

It is assumed that the two variables X_1 and X_2 are random and uniformly distributed in the interval $[0, 2]$. The goal is to estimate the mean value $\mathbb{E}[b]$ of the function b as well as the correlation coefficients $\rho(b, X_1)$ and $\rho(b, X_2)$. The exact values of these statistics can be easily computed analytically

$$\begin{aligned} \mathbb{E}[b] &= \int_0^2 \int_0^2 b(x_1, x_2) \frac{1}{4} \, \mathrm{d}x_1 \mathrm{d}x_2 = 187, \\ \mathbb{E}[X_1] &= \mathbb{E}[X_2] = 1, \end{aligned} \tag{24}$$

$$\begin{aligned} \sigma_b &= \sqrt{\int_0^2 \int_0^2 [b(x_1, x_2) - 187]^2 \, \frac{1}{4} \, \mathrm{d}x_1 \mathrm{d}x_2} = 255.55, \\ \sigma_{X_1} &= \sigma_{X_2} = \frac{2}{\sqrt{12}}, \end{aligned} \tag{25}$$

$$\rho(b, X_1) = \frac{1}{\sigma_b \sigma_{X_1}} \int_0^2 \int_0^2 \frac{[b(x_1, x_2) - 187] \, (x_1 - 1)}{4} \mathrm{d}x_1 \mathrm{d}x_2 = 0.54, \tag{26}$$

$$\rho(b, X_2) = \frac{1}{\sigma_b \sigma_{X_2}} \int_0^2 \int_0^2 \frac{[b(x_1, x_2) - 187] \, (x_2 - 1)}{4} \mathrm{d}x_1 \mathrm{d}x_2 = -0.15 \tag{27}$$

To evaluate the different sampling methods we now set out to calculate these values with the OLH, RLH and the 'standard' Monte Carlo (MC) method.

The results of the computations are shown in Tab. 1, where the average of the error percentage in the estimates is shown. To explain how these numbers were obtained let us take the example of the estimator of the mean value $\mathbb{E}[b]$. First, M designs of experiments $\{\mathbf{X}^{(k)}\}_{k=1}^{M}$ with the method in question are determined (in the computations $M = 100$ was selected). The elements of the matrices $\mathbf{X}^{(k)}$ are denoted by $x_{ij}^{(k)}$, where $i = 1, \ldots, K$ and $j = 1, 2$. Next, based on these, M estimates $\overline{m}^{(k)}$ of the mean $\mathbb{E}[b]$ are computed,

$$\overline{m}^{(k)} = \frac{1}{K} \sum_{i=1}^{K} b(x_{i1}^{(k)}, x_{i2}^{(k)}). \tag{28}$$

Sample size	$\mathbb{E}[b]$: % error			$\rho(b, X_1)$: % error			$\rho(b, X_2)$: % error		
K	OLH	RLH	MC	OLH	RLH	MC	OLH	RLH	MC
10	7.42	19.34	28.98	10.84	13.75	50.74	64.38	188.6	346.9
20	3.89	13.02	26.31	5.18	9.27	23.00	31.33	94.30	117.2
50	1.88	9.30	15.01	2.92	5.85	13.60	14.59	67.62	79.92
100	1.02	5.45	10.68	1.77	3.82	8.34	8.83	43.36	39.33
200	0.73	3.88	7.53	1.31	2.81	5.91	5.49	30.17	33.94
500	-	2.48	4.43	-	1.77	4.04	-	17.73	21.47
1000	-	1.94	3.52	-	1.40	2.46	-	14.72	15.02
2000	-	1.29	2.55	-	0.87	1.97	-	9.84	11.11
5000	-	0.83	1.48	-	0.56	1.19	-	5.76	7.33

Table 1: The average of the error percentage for the different sampling methods and sample sizes. OLH with more than 200 points have not been computed because of the long computational time.

The value given in Tab. 1 is then the average of the error of these estimates, given as a percentage of the exact mean value,

$$\frac{1}{187M}\sum_{k=1}^{M}|\overline{m}^{(k)} - 187|. \tag{29}$$

By examining the results in the table it is easy to rank the methods. The OLH with 50 samples points gives equally good results as Monte Carlo with 1000 points and OLH with only 20 points provides the better estimations than 200 points MC simulation. On the other hand, sampling based on the RLH design does not seem to be that much advantageous when compared to MC. Thus, it can be seen that the OLH sampling is of the fundamental importance for the efficiency of the presented optimization algorithm. It provides a cheap and accurate estimation of the values of correlation coefficients between the constraints functions and the random variables, which are subsequently used in the algorithm to find the change 'direction'.

4 Example: Constrained minimization of the Rosenbrock function

Before applying the algorithm to a more complex realistic application problem let us first consider a simpler case of constrained minimization of the Rosenbrock function (23). The problem is formulated as follows:

$$\begin{aligned}
\text{find:}\quad & y_1 = \mu_{X_1},\ y_2 = \mu_{X_2}, && (30)\\
\text{that minimizes}\quad & f(\mathbf{y}) = 100(y_2 - y_1^2)^2 + (1 - y_1)^2, && (31)\\
\text{subject to:}\quad & \mathbb{P}[c_1(\mathbf{X})] \geq 0.99, && (32)\\
& \mathbb{P}[c_2(\mathbf{X})] \geq 0.99, && (33)\\
& \mathbb{P}[c_3(\mathbf{X})] \geq 0.99, && (34)\\
& 0 \leq y_1 \leq 3, && (35)\\
& 0 \leq y_2 \leq 3, && (36)
\end{aligned}$$

where

$$c_1(\mathbf{X}) = X_7(X_3X_1 + X_2 + X_4), \tag{37}$$
$$c_2(\mathbf{X}) = X_7(X_5X_1 - X_2 + X_6), \tag{38}$$
$$c_3(\mathbf{X}) = X_7(-2X_1 - X_2 + 7), \tag{39}$$

and $X_1, \ldots, X_7$ are random variables with the probability distributions given in Tab. 2.

	Distribution	Mean	Std. dev.
X_1	normal	y_1	0.2
X_2	normal	y_2	0.2
X_3	uniform	-0.5	0.029
X_4	uniform	0.0	0.058
X_5	uniform	2.0	0.058
X_6	uniform	-0.8	0.058
X_7	normal	1.0	0.02

Table 2: Random variables in the Rosenbrock function minimization problem

Thus, we are looking for the mean values of random variables X_1 and X_2 in the range $[0, 3]$ that would minimize the Rosenbrock function and satisfy the reliability constraints (32)–(34).

The contours of the objective function and the feasible domain enclosed by the constraints lines: $c_1(\mathbf{x}) = 0$, $c_2(\mathbf{x}) = 0$ and $c_3(\mathbf{x}) = 0$ are shown in Fig. 3. Variables $X_3, \ldots, X_6$ are the random parameters of the first two constraint functions resulting in the uncertainty of the feasible domain definition. The random factor X_7 could be interpreted as an additional 'noise' effect in the computation of constraints values. Analyzing Fig. 3, it is quite evident that the minimum point (1,1), from the deterministic optimization, is very sensitive to parameter variations and thus does not give a reliable design.

To illustrate the method, an example of design history is shown in Fig. 4. The initial design was found using MOLH sampling (see section 3.3) with 60 points and assuming uniform distribution of X_1 and X_2 in the interval $[0, 3]$ (so, projecting into 1D, the distance between adjacent two points equals 0.05). At the subsequent iteration steps, 40 points OLH sampling was used to assess the values of reliability constraints and to evaluate correlation coefficients between c_1, c_2, c_3 and variables X_1 and X_2. The scatter of points around the initial and optimal/improved designs presented in Fig. 4 demonstrates that in the presence of uncertainties the initial design is not acceptable. However, as was stressed in section 2.1 the optimal design obtained with the presented approach depends on the size of the sample used to check the reliability constraints. Since the admissible probabilities p^a in (32)–(34) were equal 0.99 and only 40 sample points were used, one can only claim that the resulting design is more reliable than the initial one but not that the reliability constraints are satisfied.

The results of the optimization process very much depend on the choice of the starting point, which on the other hand depends on the sample size. In order to analyze the scatter of the results 30 optimization processes were performed, first, using 40 points OLH sampling and next, 100 points OLH. In the first case 60 points MOLH was used to find the starting point, in the second case it was 100 points MOLH. It was decided that the optimization process is stopped after the first feasible design (i.e. 'satisfying' all reliability constraints) is found. To speed up the computations the weight factor w_f in criterion (11) was set to zero (this strategy was mentioned in section 2.2.2).

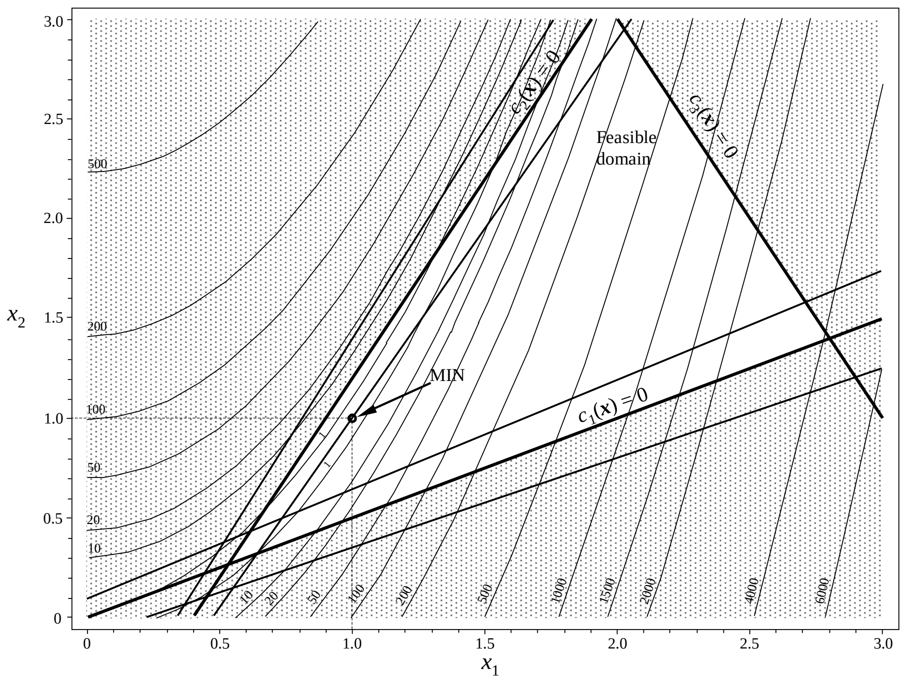

Figure 3: Rosenbrock function contours and the constraints of the optimization problem. The uncertainty of the parameters defining constraints $c_1(\mathbf{X})$ and $c_2(\mathbf{X})$ results in the feasible domain uncertainty.

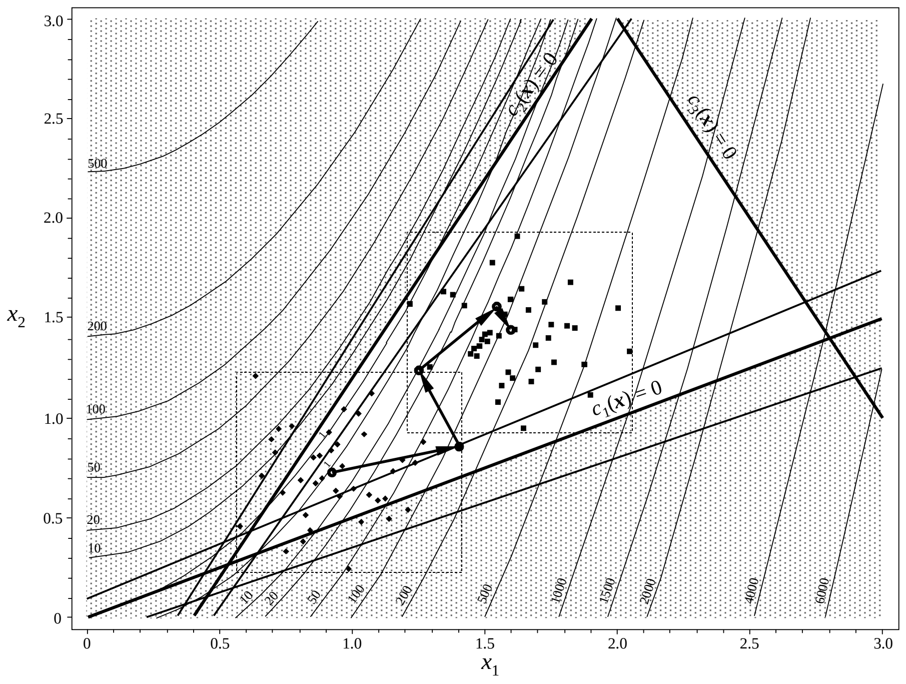

Figure 4: Iterations history for the case with 40 sample points. The scatters of possible designs around the initial and final points are shown.

In Tab. 3, the mean values and the standard deviations of the design variables and the objective function

	y_1		y_2		f	
	Mean	Std.dev.	Mean	Std.dev.	Mean	Std.dev.
OLH 40	1.71	0.13	1.50	0.17	213.6	104.2
OLH 100	1.69	0.10	1.47	0.12	200.2	73.2
RLH 15000	1.68	0.03	1.47	0.07	186.6	14.7

Table 3: Results of the Rosenbrock function minimization problem for different sampling methods.

are presented. These results are compared with the corresponding values obtained using RLH sampling technique with 15000 sample points. Because of the large sample size and the small scatter of results, these values could be treated as a reference. It is interesting to observe comparing the OLH 40 and the OLH 100 results that the mean values of y_1, y_2 and f do not differ significantly. However, the scatter of results is smaller in the case of the optimization with the 100 points OLH sampling. This can also be seen in Fig. 5 where the optimal design points are shown. The 'envelops' enclose points resulting from the same sampling method, shaded region corresponds to the RLH results. Of course, such a comparison can only be performed in a simple case when the objective and the constraints are explicitly given.

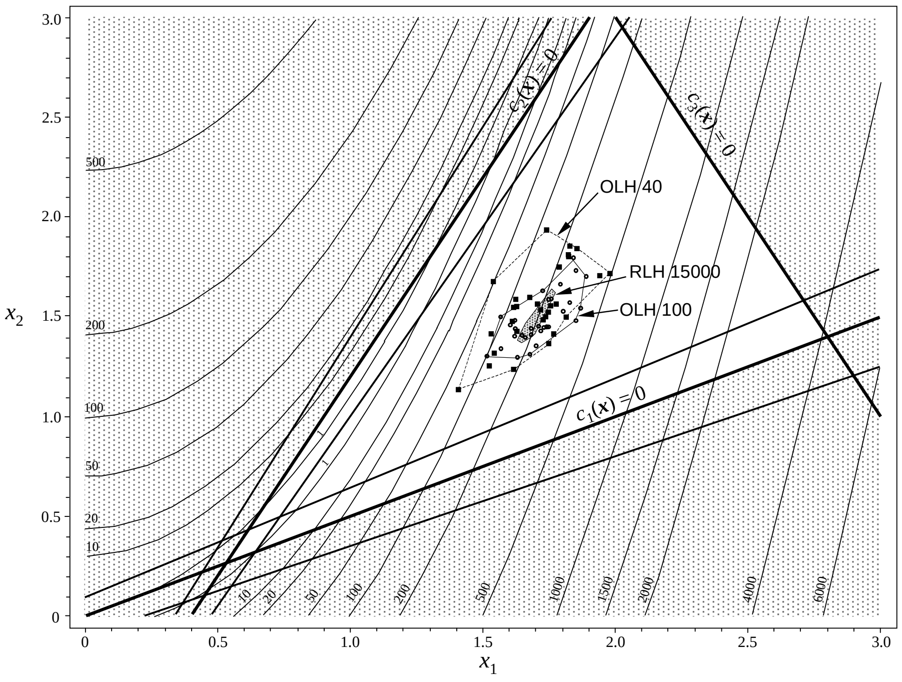

Figure 5: Scatter of the optimal points for various sampling techniques.

5 Example: A crash of the thin-walled s-beam with uncertain parameters

Here we consider a problem of the thin-walled steel s-beam, shown in Fig. 6, clamped at one end and hit at the other end by the 100kg mass moving with the initial velocity $v_0 = 15\mathrm{m/s}$ in the x-axis direction. The beam consists of 3 omega-shaped parts and the cover plate. The omega parts are attached to the

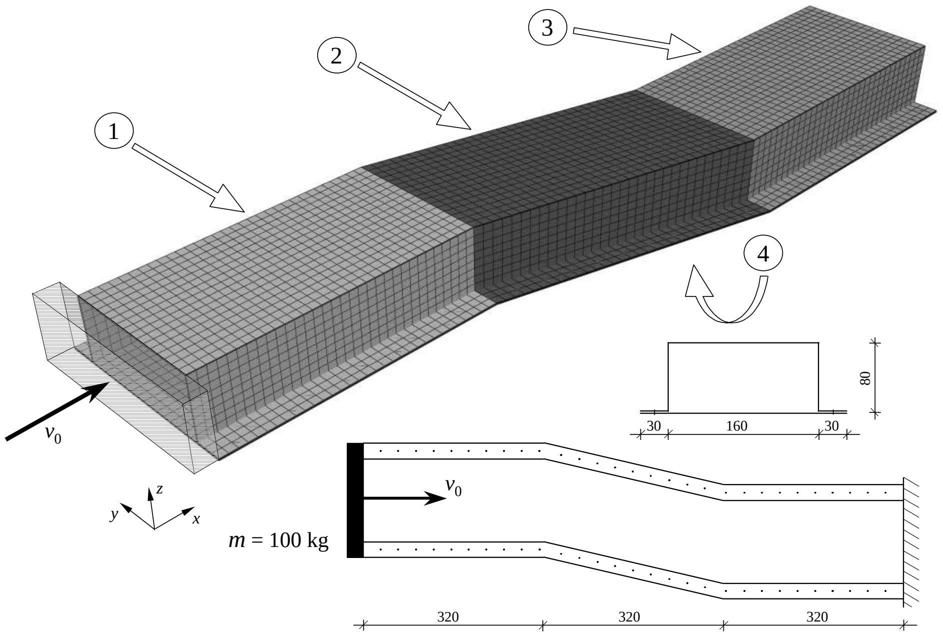

Figure 6: S-beam crash problem. The finite element model and geometry. Dimensions are in millimeters. The arrows indicate the locations of the parts.

cover with 64 spotwelds. The finite element models consists of 5760 MITC4 type shell elements [5] and 64 spring elements to model the spotwelds. The finite element analysis was performed using the explicit finite element software [12], particularly developed for the analysis of highly dynamic and nonlinear problems, in particular crash.

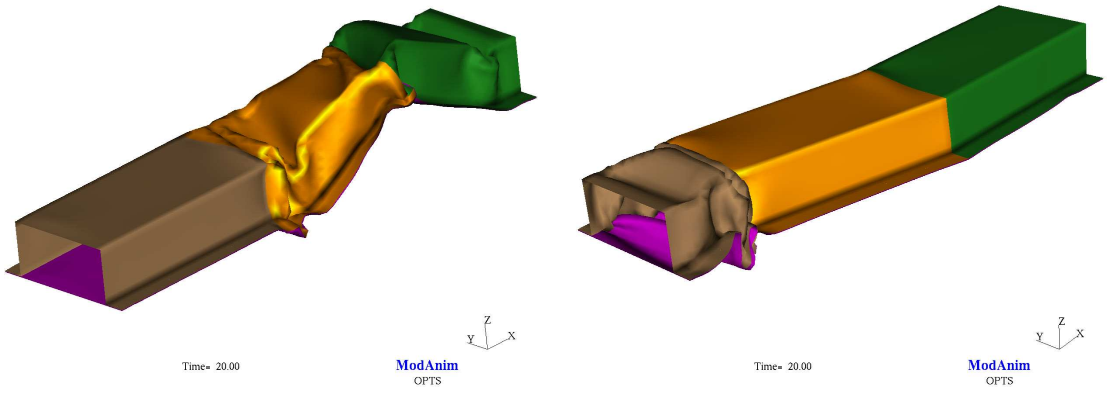

Figure 7: left: buckling type deformation, little absorbed energy; right: regular folding, good energy management

The beam acts here as an energy absorbing device. The major concern in its design is to ensure that it guarantees a good energy management by collapsing in regular folding rather than buckling mode (see Fig. 7). However, very often a design that performs satisfactorily in the ideal (nominal) operating

	Description	Distribution	Mean	Std. dev./c.o.v.
X_1	t_1 - thickness of the part 1	lognormal	y_1 [mm]	ν_{X_1} = 5%
X_2	t_2 - thickness of the part 2	lognormal	y_2 [mm]	ν_{X_2} = 5%
X_3	t_3 - thickness of the part 3	lognormal	y_3 [mm]	ν_{X_3} = 5%
X_4	t_4 - thickness of the part 4	lognormal	y_4 [mm]	ν_{X_4} = 5%
X_5	σ_0 - yield stress	lognormal	180 [MPa]	σ_{X_5} = 15 [MPa]
X_6	$\sigma_{\max}$ - maximum stress	lognormal	350 [MPa]	σ_{X_6} = 15 [MPa]
X_7	v_0^y - y component of the mass initial velocity	normal	0 [m/s]	σ_{X_7} = 1.5 [m/s]
X_8	v_0^z - z component of the mass initial velocity	normal	0 [m/s]	σ_{X_8} = 1.5 [m/s]

Table 4: Random variables in the s-beam crash problem

conditions is not reliable due to unavoidable uncertainties of some parameters. In reality it is hard to guarantee that the mass will impact the beam precisely in the assumed direction, that all the spotwelds are well made and the thicknesses of the metal parts do not differ from their nominal values. For this reasons it seems essential that a design is verified for its sensitivity to parameter uncertainties and, if necessary, improved to satisfy certain reliability level. Since the presented optimization/improvement method is based on the scatter analysis of system's responses it is important to carefully check if the finite element model is good enough to represent the physical phenomenon and is not the source of the scatter of the results. If a slight change in the FE mesh or different choice of contact algorithm have similar influence as uncertainties of physical parameters, the model should not be used for stochastic analysis.

In our problem 8 random variables were identified. Their description is presented in Tab. 4. Variables X_5 and X_6 are the parameters of the Johnson Cook elastic plastic brittle material law [7] while the variables X_7 and X_8 account for variations of the initial conditions. To account for uncertain quality of the spotweld connections 3 ($\approx$ 5%) randomly selected spring elements are being deleted from the model. The mean values of the first four random variables (thicknesses of the parts) are chosen as design variables. Since their values change in the optimization process the corresponding standard deviations will also change in order to keep the coefficient of variation constant and equal to 5%.

We formulate the optimization problem as follows:

$$\text{find } \mathbf{y} = \{y_1 = \mu_{X_1}, y_2 = \mu_{X_2}, y_3 = \mu_{X_3}, y_4 = \mu_{X_4}\}, \tag{40}$$

$$\text{that minimizes volume of material } V(\mathbf{y}), \tag{41}$$

$$\text{subject to: for all the sample points:}$$

$$1) \text{ absorbed energy is greater then } 7500\,\text{J}, \tag{42}$$

$$2) \text{ deformation of the part 1 is greater than 50\% of the total deformation}, \tag{43}$$

$$0.8\text{mm} \le y_i \le 2.0\text{mm}, \qquad i = 1, \dots, 4. \tag{44}$$

The crash duration is taken as 20ms. The minimal admissible value of absorbed energy in the constraint (42) corresponds to the design $t_1 = t_2 = t_3 = 1.5$mm, $t_4 = 1$mm and the mean values of the variables $X_5 \dots X_8$. The constraint (43) was introduced to favor the designs which lead to deformation localized in part 1.

The initial design was selected using MOLH technique (see sections 2.2.1 and 3.3) with 50 sample points. In Fig. 8, where the design history is shown, the starting point corresponds to iteration 0. It can be seen in Fig. 9 that this initial design is not reliable. 28 out of 50 sample points do not satisfy the first constraint and 1 does not satisfy the second one. Similarly to the first example, the adopted strategy was to find the first acceptable design using 50 points OLH sampling. Such a design was found in 4 iterations. As seen in Fig. 8, the volume of the beam must be increased with the thicknesses of parts 2 and 3 being almost equal and close to the upper bound and the thickness of the part 1 considerably smaller.

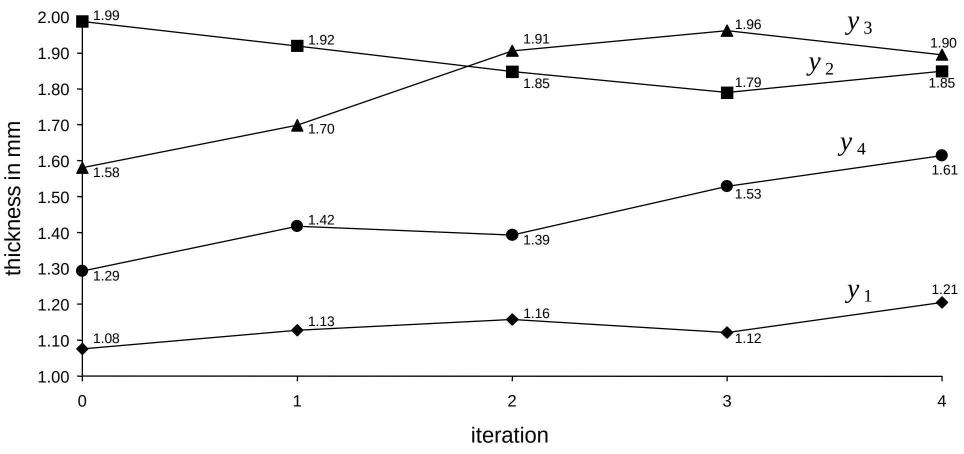

Figure 8: Design history

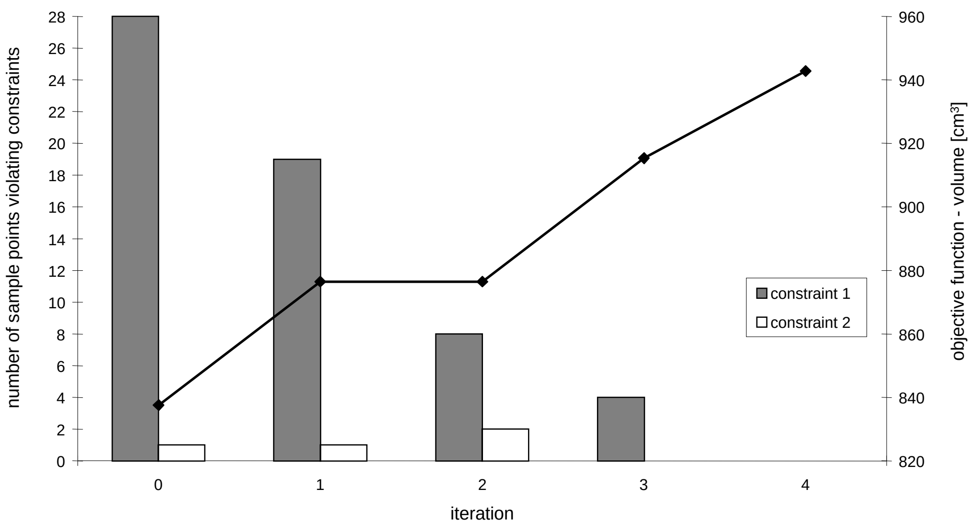

Figure 9: Objective function and constraints history

6 Conclusions

It is often the case that the design that performs satisfactorily for nominal values of its parameters is not reliable. This means that the unavoidable imperfections of geometrical or material parameters as well as loading and initial conditions may lead to an unwanted behaviour.

The reliability improvement method that was proposed in this paper is based on stochastic simulations with the samples generated according to the OLH designs. It was shown that OLH sampling is an efficient method to assess the statistical moments of the functions of random variables. The proposed algorithm for improvement makes use of correlations between the design constraint functions and the random system parameters. Thanks to the adopted sampling technique moderate size samples are able to produce accurate estimations of correlation coefficients. Since for a large number of random variables and many sample points the OLH generation may take a long time, two OLH generation algorithms were tested. It was concluded that the CP algorithm is more efficient for small and medium size Latin hypercubes while the genetic algorithm is better suited for large OLH designs.

For computationally expensive problems, like e.g. crash analysis, when only limited number of simulations can be afforded and when the design sensitivities are either not available or very inaccurate the presented method seems to be an acceptable solution. It is also the case when the response surface models of the phenomenon oversimplify the real responses and their use together with the standard RBO methods is likely to produce substantial errors. However, it must be realized that the method is best suited for problems where the probability of failure for the initial design is large and the main interest is to find more reliable design rather than the optimal one in the sense of RBO.

Acknowledgments
The financial supported from The Foundation of Polish Science (FNP nr 4/2001) is gratefully acknowledged.

References

[1] P. Audze and V. Eglais. New approach to planning out of experiments. In *Problems of dynamics and strength*, volume 35, pages 104–107, 1977. (in Russian).

[2] M. Avalle, G. Chiandussi, and G. Belingardi. Design optimization by response surface methodology: application to crashworthiness design of vehicle structure. *Struct Multidisc Optim*, 24:325–332, 2002.

[3] V. Braibant, M. Bulik, M. Liefvendahl, S. Molinier, R. Stocki, and C. Wauquiez. Stochastic simulation of highly nonlinear dynamic systems using the M-XPLORE extension of the RADIOSS software. In *Proc. of Workshop on Optimal Design of Materials and Structures, Ecole Polytechnique, Palaiseau, France, 25-27 November*, 2002. On CD.

[4] K.J. Craig, N. Stander, D Dooge, and S. Varadappa. MDO of automotive vehicle for crashworthiness and NVH using response surface methods. In *Proc. 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, 4–6 September*, 2002. Paper 2002-5607.

[5] E.N. Dvorkin and K.J. Bathe. A continuum mechanics based four-node shell element for general nonlinear analysis. *Eng Comput*, 1:77–88, 1984.

[6] L. Gu and R.J. Yang. Recent applications on reliability-based optimization of automotive structures. In *Reliability & Robust Design in Automotive Engineering*, pages 77–88. SAE, 2003.

[7] G. Johnson and W. Cook. A constitutive model and data for metals subjected to large stains, high strain rates and high temperatures. In *7th Symposium on Ballistics, The Hague*, 1983.

[8] N. Kuschel and R. Rackwitz. Two basic problems in reliability-based structural optimization. *Mathematical Methods of Operations Research*, 46:309–333, 1997.

[9] M. Liefvendahl and R. Stocki. A study on algorithms for optimization of Latin hypercubes. submitted for reviews in Journal of Statistical Planning and Inference, 2003.

[10] H.O. Madsen and P. Friis Hansen. A comparison of some algorithms for reliability based structural optimization and sensitivity analysis. In R. Rackwitz and P. Thoft-Christensen, editors, *Reliability and Optimization of Structural Systems '91, Proc. 4th IFIP WG 7.5 Conf., Munich, 11–13 September 1991*, pages 443–451. Springer-Verlag, 1992.

[11] J. Marczyk. Stochastic multidisciplinary improvement: beyond optimization. In *Proc. 8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, September*, 2000.

[12] Mecalog SARL, 2 Rue de la Renaissance 92160 Antony, France. *RADIOSS Input Manual, Version 4.2*, 2000.

[13] R.E. Melchers. *Structural Reliability Analysis and Predictions, 2nd Ed.* Wiley, 1999.

[14] T.J. Mitchell. Computer construction of D-optimal first-order designs. *Technometrics*, 16:211–220, 1974.

[15] A. Nataf. Determination des distribution dont les marges sont données. *Comptes Rendus de l'Academie des Sciences*, 225:42–43, 1962.

[16] A.M.J. Olsson, G.E. Sandberg, and O. Dahlblom. Latin hypercube sampling - a tool in structural reliability analysis. In *Proc. ECCM-2001, Cracow, Poland, 26–29 June*, 2001. on CD.

[17] J.-S. Park. Optimal latin-hypercube designs for computer experiments. *Journal of statistical planning and inference*, 39:95–111, 1994.

[18] M. Redhe, J. Forsberg, T. Jansson, P-O. Marklund, and L. Nilsson. Using the response surface methodology and the d-optimality criterion in crashworthiness related problems. *Struct Multidisc Optim*, 24:185–194, 2002.

[19] M. Rossenblatt. Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23:470–472, 1952.

[20] T.W. Simpson. *A concept exploration method for product family design*. PhD thesis, Georgia Inst. of Tech., 1998.

[21] J. Sobieszczanski-Sobieski, S. Kodiyalam, and R.J. Yang. Optimization of car body under constraints of noise, vibration, and harshness (NVH), and crash. *Struct Multidisc Optim*, 22:295–306, 2001.

[22] *ST-ORM, A Meta-Computing System for Stochastic Optimization and Robustness Management*. EASi Engineering GmbH, Germany, 2000.

[23] N. Stander, W. Roux, M. Giger, M. Redhe, N. Fedorova, and J. Haarhoff. Crashworthiness optimization in LS-OPT: case studies in metamodelling and random search techniques. In *Proc. 4th European LS-DYNA Users Conference, Ulm, 22–23 May*, 2003.

[24] K.Q. Ye, W. Li, and A. Sudjianto. Algorithmic construction of optimal symmetric Latin hypercubes. *Journal of statistical planning and inference*, 90:145–159, 2000.

[25] T. Zou, S. Mahadevan, and Z. Mourelatos. Efficient reliability methods for automotive structural systems. In *Proc. The 15th Engineering Mechanics Conference of ASCE, New York, NY*, 2002.