TILBURG UNIVERSITY

This is a postprint of:

Space-filling Latin hypercube designs for computer experiments

E.R. van Dam, D. den Hertog, B.G.M. Husslage & G. Rennen

Optimization and Engineering, 2011, vol. 12, nr. 4, pp. 611-630

For citation:

Please use the reference above

Link to the postprint file:

http://arno.uvt.nl/show.cgi?fid=120516

More information about this publication:

http://repository.uvt.nl/id/ir-uvt-nl:oai:wo.uvt.nl:4334874

Optim Eng (2011) 12:611–630
DOI 10.1007/s11081-010-9129-8

Space-filling Latin hypercube designs for computer experiments

Bart G.M. Husslage · Gijs Rennen · Edwin R. van Dam · Dick den Hertog

Received: 5 February 2008 / Accepted: 18 November 2010 / Published online: 26 November 2010

Abstract In the area of computer simulation, Latin hypercube designs play an important role. In this paper the classes of maximin and Audze-Eglais Latin hypercube designs are considered. Up to now only several two-dimensional designs and a few higher dimensional designs for these classes have been published. Using periodic designs and the Enhanced Stochastic Evolutionary algorithm of Jin et al. (J. Stat. Plan. Interference 134(1):268–687, 2005), we obtain new results which we compare to existing results. We thus construct a database of approximate maximin and Audze-Eglais Latin hypercube designs for up to ten dimensions and for up to 300 design points. All these designs can be downloaded from the website http://www.spacefillingdesigns.nl.

Keywords Audze-Eglais · Computer experiment · Enhanced stochastic evolutionary algorithm · Latin hypercube design · Maximin · Non-collapsing · Packing problem · Simulated annealing · Space-filling

This paper is a revision of Husslage et al. (2006).
The research of B.G.M. Husslage has been financially supported by the SamenwerkingsOrgaan Brabantse Universiteiten (SOBU).
The research of E.R. van Dam has been made possible by a fellowship of the Royal Netherlands Academy of Arts and Sciences.

B.G.M. Husslage · G. Rennen (✉) · E.R. van Dam · D. den Hertog
Department of Econometrics and Operations Research, Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands
e-mail: g.rennen@gmail.com

B.G.M. Husslage
e-mail: b.g.m.husslage@uvt.nl

E.R. van Dam
e-mail: edwin.vandam@uvt.nl

D. den Hertog
e-mail: d.denhertog@uvt.nl

1 Introduction

A k-dimensional Latin hypercube design (LHD) of n points, is a set of n points $x_i = (x_{i1}, x_{i2}, \dots, x_{ik}) \in \{0, \dots, n-1\}^k$ such that for each dimension j all x_{ij} are distinct. In this definition, we assume that our design space is equal to the $[0; n-1]^k$ hypercube. However by scaling, we can use LHDs for any rectangular design space. Alternative definitions of LHDs also occur in the literature. One alternative definition is to divide each axis into n equally sized bins and randomly select points such that each bin contains exactly one point. However, we refer to this technique as Latin hypercube sampling (LHS). In this paper the term 'LHD' thus only refers to the first definition.

An LHD is called maximin when the separation distance $\min_{i \neq j} d(x_i, x_j)$ is maximal among all LHDs of given size n, where d is a certain distance measure. In this paper, we concentrate on the Euclidean (or ℓ^2) distance measure, i.e.,

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^{k} (x_{il} - x_{jl})^2}, \tag{1}$$

since this measure is often the first choice in practice.

Besides maximin LHDs, we also treat Audze-Eglais LHDs. These LHDs minimize the following objective:

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{1}{d(x_i, x_j)^2}, \tag{2}$$

where $d(x_i, x_j)$ is again the Euclidean distance between points x_i and x_j. By minimizing this objective, we can also obtain LHDs with "evenly spread" points (Bates et al. 2004).

For both classes of LHDs, we aim to construct a database of the best designs known in literature. We do this by generating new designs and comparing them with existing results. These designs are often approximate maximin or Audze-Eglais designs in the sense that optimality of the objective is not guaranteed. The reason for this is that optimization over the total set of LHDs can be very time-consuming for larger values of k and n. Therefore, in order to find good designs, optimization is often done over a certain class of LHDs or heuristics are used which do not guarantee optimality. The periodic LHDs described in this paper are a good example of the first case. Examples of the second case are simulated annealing used by Morris and Mitchell (1995), the permutation genetic algorithm of Bates et al. (2004) and the Enhanced Stochastic Evolutionary (ESE) algorithm of Jin et al. (2005).

The designs which are best according to the comparison in this paper are added to the website http://www.spacefillingdesigns.nl where they can be downloaded for free. As far as we know this is the first extensive online catalogue of maximin and Audze-Eglais LHDs, although there are several catalogues for classical design of experiments, see e.g., the WebDOE™ website of Crary (2008). Crary et al. (2000) developed I-OPT™ to generate designs with minimal integrated mean squared error (IMSE). They found that IMSE-optimal designs can have proximate design points, which they call "twin points"; see also Crary (2002).

Our main motivation for investigating this subject is that maximin and Audze-Eglais Latin hypercube designs are very useful in the area of computer simulation. One important area where computer simulation is used a lot is engineering. Engineers are confronted with the task of designing products and processes. Since physical experimentation is often expensive and difficult, computer models are frequently used for simulating physical characteristics. The engineer often needs to optimize the product or process design, i.e., to find the best settings for a number of design parameters that influence the critical quality characteristics of the product or process. A computer simulation run is usually time-consuming and there is a great variety of possible input combinations. For these reasons, meta-models that model the quality characteristics as explicit functions of the design parameters are constructed. Such a meta-model, also called a (global) approximation model or surrogate model, is obtained by simulating a number of design points. Well-known meta-model types are polynomials and Kriging models. Since a meta-model evaluation is much faster than a simulation run, in practice such a meta-model is used, instead of the simulation model, to gain insight into the characteristics of the product or process and to optimize it. A review of meta-modeling applications in structural optimization can be found in Barthelemy and Haftka (1993), and in multidisciplinary design optimization in Sobieszczanski-Sobieski and Haftka (1997).

As observed by many researchers, there is an important distinction between designs for computer experiments and designs for the more traditional response surface methods. Physical experiments exhibit random errors and computer experiments are often deterministic (cf. Simpson et al. 2004). This distinction is crucial and much research is therefore aimed at obtaining efficient designs for computer experiments.

As is recognized by several authors, such a design for computer experiments should at least satisfy the following two criteria (see Johnson et al. 1990, Morris and Mitchell 1995, and Simpson et al. 2001). First of all, the design should be *space-filling* in some sense. When no details on the functional behavior of the response parameters are available, it is important to be able to obtain information from the entire design space. Therefore, design points should be "evenly spread" over the entire region. One of the measures often used to obtain space-filling designs is the maximin measure (see p. 148 of Santner et al. 2003 and p. 17 of Forrester et al. 2006). The Audze-Eglais measure is another measure used for this purpose. Secondly, the design should be *non-collapsing*. When one of the design parameters has (almost) no influence on the function value, two design points that differ only in this parameter will "collapse", i.e., they can be considered as the same point that is evaluated twice. For deterministic simulation models this is not a desirable situation. Therefore, two design points should not share any coordinate values when it is not known a priori which dimensions are important. Note that in other fields of research such designs are referred to as *low discrepancy* designs. To obtain non-collapsing designs the Latin hypercube structure is often enforced. It can be shown that if the function of interest is independent of one or more of the k parameters then, after removal of the irrelevant parameters, the projection of the LHD onto the reduced design space retains good spatial properties; see Koehler and Owen (1996). Maximin LHDs are frequently used in practical applications, see e.g., the examples given in Driessen et al. (2002), den Hertog and Stehouwer (2002), Alam et al. (2004), and Rikards and Auzins (2004).

Only a few authors consider the construction of maximin LHDs. For example, Morris and Mitchell (1995) used simulated annealing to find approximate maximin LHDs for up to five dimensions and up to 12 design points, and a few larger values, with respect to the ℓ^1- and ℓ^2-distance measure. van Dam et al. (2007) derived general formulas for two-dimensional maximin LHDs, when the distance measure is ℓ^∞ or ℓ^1, while for the ℓ^2-distance measure (approximate) maximin LHDs up to 1000 design points were obtained by using a branch-and-bound algorithm and constructing (adapted) periodic designs. Ye et al. (2000) proposed an exchange algorithm for finding approximate maximin symmetric LHDs. The symmetry property is used as a compromise between computing effort and design optimality. Jin et al. (2005) described an enhanced stochastic evolutionary (ESE) algorithm for finding approximate maximin LHDs. They also apply their method to other space-filling criteria. The Statistics Toolbox of Matlab also contains a function `lhsdesign` to generate approximate maximin LHDs. This function randomly generates a number of LHDs and picks the one with the largest separation distance. Although this method is very fast, other methods generally result in much better space-filling LHDs. To asses the quality of approximate maximin LHDs, van Dam et al. (2009) generated upper bounds on the separation distance for certain classes of maximin LHDs. By comparing the separation distances of LHDs to these bounds, we can get an indication of their quality.

There is much more literature related to maximin designs that are not restricted to LHDs. Note that a maximin design is certainly space-filling, but not necessarily non-collapsing.

First of all, the problem of finding the maximal common radius of n circles which can be packed into a square is equivalent to the maximin design problem in two dimensions. Melissen (1997) gives a comprehensive overview of the historical developments and state-of-the-art research in this field. For the ℓ^2-distance measure in the two-dimensional case, optimal solutions are known for $n \leq 30$ and $n = 36$, see e.g., Kirchner and Wengerodt (1987), Peikert et al. (1991), Nurmela and Östergård (1999), and Markót and Csendes (2005). Furthermore, many good approximating solutions have been found for $n \geq 31$; see the Packomania website of Specht (2008). Baer (1992) solved the maximum ℓ^∞-circle packing problem in a k-dimensional unit cube. The ℓ^1-circle packing problem in a square has been solved for many values of n; see Fejes Tóth (1971) and Florian (1989).

Secondly, the maximin design problem has been studied in location theory. In this area of research, the problem is usually referred to as the *max-min facility dispersion problem* (see Erkut 1990). Facilities are placed such that the minimal distance to any other facility is maximal. Again, the resulting solution is certainly space-filling, but not necessarily non-collapsing. A few papers consider maximin designs in higher dimensions, e.g., Trosset (1999), Locatelli and Raber (2002), Stinstra et al. (2003), and Dimnaku et al. (2005). These papers describe nonlinear programming heuristics to find approximate maximin designs. In most papers, a rectangular design space is assumed, but Trosset (1999), Stinstra et al. (2003) and Dimnaku et al. (2005) also specifically consider design spaces with different shapes.

Audze-Eglais LHDs are also constructed by only a few authors. The criterion was first introduced by Audze and Eglais (1977) and is based on the analogy of minimizing forces between charged particles. In Bates et al. (2004), the problem of finding

Audze-Eglais LHDs is formulated and a permutation genetic algorithm is used to generate them. Liefvendahl and Stocki (2006) compared maximin and Audze-Eglais LHDs and recommend the Audze-Eglais criterion over the maximin criterion. Examples of practical applications of Audze-Eglais LHDs can be found in Rikards et al. (2001), Bulik et al. (2004), Stocki (2005), and Hino et al. (2006).

There are several other measures proposed in the literature besides maximin and Audze-Eglais, e.g., maximum entropy, minimax, IMSE, and discrepancy. For a good overview, we refer to Koehler and Owen (1996). In statistical environments, Latin hypercube sampling (LHS) is often used. In such an approach, points on the grid are sampled without replacement, thereby deriving a random permutation for each dimension; see McKay et al. (1979). Giunta et al. (2003) give an overview of pseudo- and quasi-Monte Carlo sampling, LHS, orthogonal array sampling, and Hammersley sequence sampling. They notice that the basic LHS technique can lead to designs with poor space-filling properties. Extensions to the basic LHS technique are therefore necessary to obtain better designs but these are unfortunately not standard yet in all software packages. Bates et al. (1996) obtained designs for computer experiments by exploring so-called lattice points and using results from number theory.

Several papers combine space-filling criteria with the Latin hypercube structure. Jin et al. (2005) described an enhanced stochastic evolutionary algorithm for finding maximum entropy and uniform designs. van Dam (2008) derived interesting results for two-dimensional minimax LHDs. Rennen et al. (2010) consider nested maximin LHDs which consist of two separate designs, one being a subset of the other.

This paper is organized as follows. Section 2 describes how periodic designs can be used to obtain good approximate maximin and Audze-Eglais LHDs. In Sect. 3, we shortly describe some heuristics found in literature used for this same purpose. The ESE-algorithm of Jin et al. (2005) described in this section and periodic designs are used to generate new approximate maximin and Audze-Eglais LHDs. Computational results for up to ten dimensions and for up to 300 design points, as well as a comparison of the new and existing results, are provided in Sect. 4. Finally, Sect. 5 contains conclusions.

2 Periodic designs

Van Dam et al. (2007) showed that two-dimensional maximin Latin hypercube designs often have a nice, periodic structure. By constructing (adapted) periodic designs, many maximin LHDs and, otherwise, good LHDs, are found for up to 1000 points. Therefore, extending this idea to higher dimensions seems natural.

Let a k-dimensional Latin hypercube design of n points be represented by the sequences $y_1, \dots, y_k$, with every y_i a permutation of the set $\{0, \dots, n-1\}$. As in the two-dimensional case, a design is constructed by fixing the first dimension, without loss of generality, to the sequence $y_1 = (0, \dots, n-1)$ and assigning (adapted) periodic sequences to all other dimensions. Two types of periodic sequences are considered. The first one is the sequence $(v_0, \dots, v_{n-1})$, where

$$v_i = (i+1)p \bmod (n+1) - 1, \quad \text{for } i = 0, \dots, n-1. \tag{3}$$

Here, p is the period of the sequence, which is chosen such that $n+1$ and p have no common divisor, i.e., $\gcd(n+1, p) = 1$, resulting in a permutation of the set $\{0, \ldots, n-1\}$.

Note that the periodic designs obtained in this way resemble *lattices*; see e.g., Bates et al. (1996). The main difference is that lattices are infinite sets of points, which may collapse, and, hence, to construct a (finite) Latin hypercube design a proper subset of non-collapsing lattice points should be chosen. For given n, the structure of the lattice will, however, not always lead to a Latin hypercube design with a sufficient number of points. This is in contrast to periodic designs, for which the modulo-operator insures that for every combination of periods p_j, with $\gcd(n+1, p_j) = 1$, $j = 2, \ldots, k$, a feasible Latin hypercube design is obtained.

The second type of sequence that is considered is the more general sequence $(w_0, \ldots, w_{n-1})$, where $w_i = (s + ip) \bmod n$ (note that we changed the modulus), for $i = 0, \ldots, n-1$. In this case, all starting points $s = 0, \ldots, p$ and all periods $p = 1, \ldots, \lfloor \frac{n}{2} \rfloor$ will be considered. Note, however, that the resulting sequence w may no longer be one-to-one, i.e., some values may occur more than once, and, hence, the resulting design may no longer be an LHD. Now, let $r > 0$ be the smallest value for which $w_r = w_0$; it then follows that $r = \frac{n}{\gcd(n,p)}$. When $r < n$ a way to construct a one-to-one sequence of length n is by shifting parts of the sequence by, say, q, and repeating this when necessary. To formulate this more explicitly, for the updated sequence w it now holds that

$$w_i = (s + ip + jq) \bmod n,$$
$$\text{for } i = jr, \ldots, (j+1)r - 1, \quad \text{and} \quad j = 0, \ldots, \gcd(n, p) - 1. \tag{4}$$

Let m represent the modulus and, hence, the type of sequence used, i.e., $m = n + 1$ corresponds to the first type and $m = n$ to the second. For given n, we now have to set the parameters (p, q, s, m) for every sequence $y_2, \ldots, y_k$.

To find the best settings for the parameters it would be best to test all values. However, when the dimension and the number of points increase the number of possibilities increases rapidly. Hence, computing all possibilities gets very time-consuming or even impossible. Therefore, three classes of parameter settings (named A, B, and C) are distinguished. The largest one, class A, consists of checking the following parameter values: $p = 1, \ldots, \lfloor \frac{n}{2} \rfloor$, $q = 1 - p, \ldots, p - 1$, $s = 0, \ldots, p$, and $m \in \{n, n+1\}$. Testing in three and four dimensions indicated that almost all adapted periodic maximin designs are based on a shift of $1 - p$, -1, or 1 (as was the case for two dimensions; see van Dam et al. (2007)). Furthermore, most maximin designs are found to have a starting point equal to either $p - 1$ or p. Class B is therefore set up to be a subset of class A with the aforementioned restrictions on the parameters q and s. Finally, for the dimensions 5 to 7 the number of possibilities has to be reduced even further, leading to parameter class C, which (based on some more test results) restricts class B to the values $q = 1$ and $s = p$, leaving the other parameters unchanged. Table 1 shows the different classes used in the computations of the approximate maximin LHDs for each dimension. For the approximate Audze-Eglais LHDs only class C is used.

As an example, consider a three-dimensional adapted periodic LHD of 22 points. For the maximin criterion, a best parameter setting (class A) is found to

Table 1 Different classes of periodic sequences are checked to generate maximin designs for each dimension

Dimension	Class A	Class B	Class C
3	$2 \leq n \leq 70$	$71 \leq n \leq 100$	–
4	$2 \leq n \leq 25$	$26 \leq n \leq 100$	–
5	–	$2 \leq n \leq 80$	$81 \leq n \leq 100$
6	–	$2 \leq n \leq 35$	$36 \leq n \leq 100$
7	–	–	$2 \leq n \leq 100$

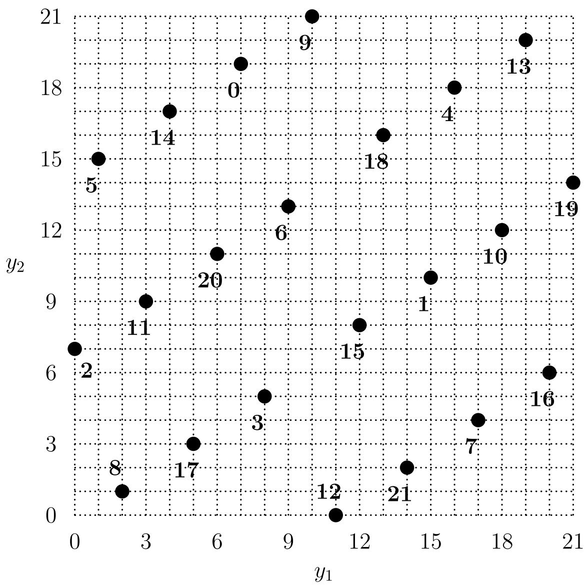

Fig. 1 Two-dimensional projection of the three-dimensional LHD (y_1, y_2, y_3) of 22 points

be $(p_2, q_2, s_2, m_2) = (8, -7, 7, 22)$ and $(p_3, q_3, s_3, m_3) = (3, 0, 2, 23)$ and, hence, the corresponding maximin LHD, with separation distance 69, is defined by the sequences

$$
\begin{aligned}
y_1 &= (0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10,\ 11,\ 12,\ 13,\ 14,\ 15,\ 16,\ 17,\ 18,\ 19,\ 20,\ 21),\\
y_2 &= (7,\ 15,\ 1,\ 9,\ 17,\ 3,\ 11,\ 19,\ 5,\ 13,\ 21,\ 0,\ 8,\ 16,\ 2,\ 10,\ 18,\ 4,\ 12,\ 20,\ 6,\ 14),\\
y_3 &= (2,\ 5,\ 8,\ 11,\ 14,\ 17,\ 20,\ 0,\ 3,\ 6,\ 9,\ 12,\ 15,\ 18,\ 21,\ 1,\ 4,\ 7,\ 10,\ 13,\ 16,\ 19).
\end{aligned} \tag{5}
$$

Thus, y_3 is a periodic sequence, with $m = n + 1$, and y_2 is an adapted periodic sequence, with $m = n$ and $q_2 = -7$. Note that to obtain a one-to-one sequence, the second part of y_2, i.e., $(0, 8, \dots, 14)$, is formed by shifting the first part of y_2, i.e., $(7, 15, \dots, 21)$, by -7. The periods and shift are clearly visible in the two-dimensional projection of the LHD in Fig. 1. In this figure the y_3-values are depicted at the design points.

Like in the two-dimensional case, it may happen that for a given n the corresponding maximin LHD has a separation distance that is smaller than the distance of a design of $n - 1$ points. For these n, however, better designs can usually be derived by adding an extra "corner point" to the LHD of $n-1$ points. In this way, a monotone nondecreasing sequence of separation distances was found for all dimensions.

3 Other methods

3.1 Enhanced stochastic evolutionary algorithm

Besides restricting ourselves to a certain class of LHDs, we can also generate good maximin or Audze-Eglais LHDs using heuristics. The ESE-algorithm of Jin et al. (2005) is one of the methods developed for this purpose and is used in this paper to generate new approximate maximin and Audze-Eglais LHDs.

This method starts with an initial design and tries to find better designs by iteratively changing the current design. To determine if a new design is accepted, a threshold-based acceptance criterion is used. This criterion is controlled in the outer loop of the algorithm. In the inner loop of the algorithm new designs are explored.

The inner loop explores the design space as follows. At each iteration, the algorithm creates a fixed number of new designs by exchanging two randomly chosen elements. The new design with the largest separation distance or with the smallest Audze-Eglais objective value is then compared to the current design with a threshold criterion. The criterion is such that it ensures that better designs are always accepted and that worse designs can also be accepted with a certain probability. If the new design is accepted, it replaces the current design. This process is repeated for a user defined number of iterations.

The outer loop controls the threshold value. After the inner loop is completed, the outer loop determines how much improvement is made in the inner loop. If the amount of improvement is above a certain level, the algorithm starts an improving process in which it tries to rapidly find a local optimum. It does this by lowering the threshold value and thus accepting less deteriorations in the inner loop. If too little improvement is made, an exploration process is started which is intended to escape from a local optimum. The threshold value is first rapidly increased to move away from a local optimum and later slowly decreased to find better designs after moving away. The final design of the algorithm is the best design found during all iterations of the inner loop.

For a more detailed description of the algorithm, we refer to the original paper of Jin et al. (2005). To find maximin and Audze-Eglais LHDs, we implemented the ESE-algorithm in Matlab. The parameters of the algorithm were set to the values suggested in Jin et al. (2005). The only adjustment we made to the original algorithm is in the choice of stopping criterion. Instead of stopping after a fixed number of runs of the outer loop, our criterion is to stop when in the last 1000 runs of the outer loop no improvement is made.

3.2 Simulated annealing

Another heuristic used to find maximin LHDs is simulated annealing. Morris and Mitchell (1995) were the first to apply simulated annealing for this purpose. The simulated annealing method tries to find good designs by iteratively changing a random starting design. These changes are chosen randomly from a predefined class of possible changes. For each design, these possible changes define a set of designs which are called the neighborhood of the design. Before a change is accepted, the new neighbor design obtained by applying the selected change is evaluated. If a change improves the current design, the change is always accepted. A key characteristic of simulated annealing is however that changes which result in a worse design can also be accepted. This enables simulated annealing to escape from local optima. A worse design is accepted with a probability which depends on two factors. Firstly, designs which are only slightly worse are accepted with a higher probability than design which are much worse. Secondly, the acceptance probability is changed during the course of the algorithm. Generally, worse designs are accepted with a higher probability at the beginning of the algorithm than at the end.

Besides Morris and Mitchell (1995), also Husslage (2006) used simulated annealing for finding maximin LHDs. One of the main differences between the two methods is the used objective function. Husslage (2006) directly used the separation distance of a design, whereas Morris and Mitchell (1995) used a surrogate measure ϕ_p. This measure also takes into account the number of pairs of points with a certain distance between them. By including this information, it is easier to decide which design is best if they have the same separation distance. This surrogate measure is also used by other authors like Jin et al. (2005) and Palmer and Tsui (2001).

Simulated annealing and ESE are similar in many respects. Both algorithms create new designs by changing a current design. Furthermore, both algorithms accept worse designs with a positive probability. The change of this acceptance probability in simulated annealing is similar to the change of the threshold value in the outer loop of ESE. The main difference between the two methods is that the ESE-algorithm creates several new designs and compares the best of these designs to the current design, whereas simulated annealing only creates one new design. The ESE-algorithm can thus be regarded as an enhancement of simulated annealing.

3.3 Permutation genetic algorithm

To obtain Audze-Eglais LHDs, Bates et al. (2004) used a permutation genetic algorithm. The genetic algorithm starts with generating a set of LHDs called a "population". The Audze-Eglais distance of each design in this population is then calculated. Based on these distances, a subset of designs is selected using so-called elitist and tournament selection. A new population of designs is created by applying mutation and crossover operations to the selected designs. By repeatedly selecting and creating designs, the Audze-Eglais distances of the LHDs in the population gradually increase. Results of this algorithm were reported by Bates et al. (2004) for eight different combinations of n and k. In Sect. 4, we make a comparison between these results, the designs obtained with periodic designs, and the designs obtained with ESE.

4 Computational results

Using (adapted) periodic designs and the ESE-algorithm, approximate maximin and Audze-Eglais LHDs have been obtained for the cases described in Table 2. All computations have been performed on PCs with a 2.8-GHz Pentium D processor. For the cases with $n > 100$, a limit of 6 hours was imposed on the calculation time.

Table 2 Largest values of n for which LHDs were generated using periodic designs (PD) and using the ESE-algorithm

Dimension	3	4	5	6	7	8	9	10
Maximin PD	300	300	100	100	100			
Maximin ESE	300	300	100	100	100	100	100	100
Audze-Eglais PD	100	100	100					
Audze-Eglais ESE	100	100	100	100	100	100	100	100

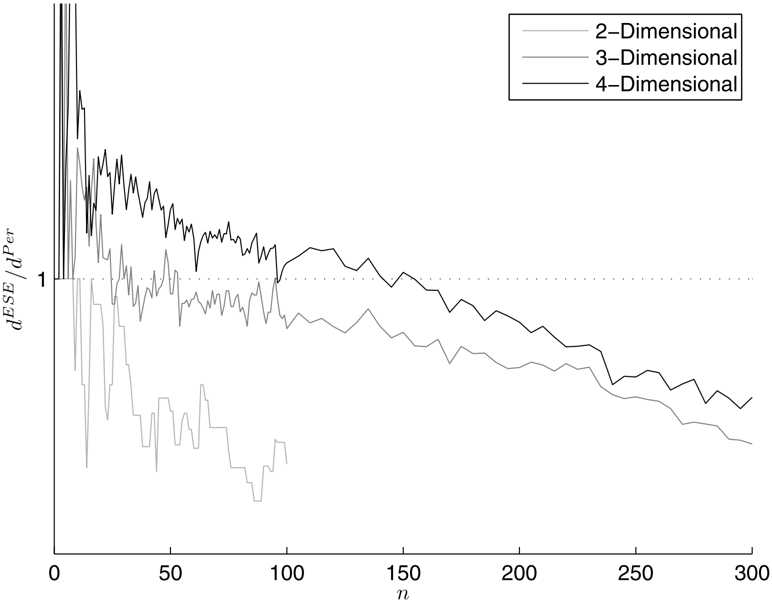

Fig. 2 Ratio between separation distance of ESE and periodic designs

Table 5 shows the squared ℓ^2-separation distance of the (approximate) maximin LHDs that were obtained by applying periodic designs and of those obtained by the ESE-algorithm. The column for two-dimensional periodic designs contains the results obtained in van Dam et al. (2007). From this table it can be seen that (adapted) periodic designs work particularly well for larger values of n and small k.

For dimension 2 to 4, Fig. 2 shows the ratio between the squared separation distance of ESE and periodic designs. A ratio larger than one indicates that the ESE design is better than the (adapted) periodic design. A break-even point, i.e., a point (or, better, an interval) where the preference shifts from the designs found by ESE to (adapted) periodic designs, is clearly visible in this figure. Furthermore, these break-even points seem to increase with the dimension of the design and it is to be expected that break-even points for k-dimensional designs, with $k \geq 5$, will occur for larger values of n, i.e., $n > 250$. Because all six- and seven-dimensional (adapted) periodic designs, of 3 to 100 points, are dominated by the designs found by ESE, the former are not computed for larger dimensions.

Table 3 Squared ℓ^2-separation distance of designs found by Morris and Mitchell (1995) vs. the ESE-algorithm

n	3 dim		4 dim		5 dim		6 dim		7 dim		8 dim		9 dim	
	M&M	ESE	M&M	ESE	M&M	ESE	M&M	ESE	M&M	ESE	M&M	ESE	M&M	ESE
3	**6**	**6**	**7**	**7**	**8**	**8**								
4	**6**	**6**	**12**	**12**	**14**	**14**								
5	**11**	**11**	**15**	**15**	**24**	**24**								
6	**14**	**14**	**22**	**22**	**32**	**32**	**40**	**40**						
7	**17**	**17**	**28**	**28**	**40**	**40**			**61**	**61**				
8	**21**	**21**	**42**	**42**	**50**	**50**					**91**	89		
9	**22**	**22**	**42**	**42**	**61**	**61**							**126**	**126**
10	**27**	**27**	**50**	47	**82**	**82**								
11	29	**30**	**55**	**55**	**80**	**80**								
12	**36**	**36**	**63**	**63**	**91**	**91**	**139**	136						
13														
14									**219**	215				

In Table 3, we compare the LHDs found by Morris and Mitchell (1995) and the ESE-algorithm. The ESE-algorithm is able to match the results of Morris and Mitchell (1995) for most combinations of k and n. Only for the cases $(k, n) = (4, 10)$, $(6, 12)$, $(7, 14)$, and $(8, 8)$ are slightly worse designs obtained. Three of these four design satisfy the property that $n = k$ or $n = 2k$. According to Morris and Mitchell (1995), these designs exhibit special symmetric properties; they refer to them as *foldover designs*. These special properties are probably the main explanation for the better results in these cases. For the case $(k, n) = (3, 11)$, we obtained an improved (and optimal) design. Furthermore, using a branch-and-bound algorithm, the three-dimensional designs of up to 15 points have been verified to be optimal (van Dam et al. 2009). From the above results, we can conclude that performances of the ESE-algorithm and the simulated annealing algorithm of Morris and Mitchell (1995) are closely matched. However, the numerical results of Morris and Mitchell (1995) are probably too limited to be useful in most practical applications.

We also compared the ESE results with the SA results in Husslage (2006) and saw that the ESE-algorithm gives better or equally good results for most combination of k and n. For only nine combinations the results are better of the SA algorithm and for 7 percent of the compared combinations the results are equally good. However, especially for larger values of n, the ESE algorithm found many designs with a more than 15 percent higher separation distance.

The results obtained for the Audze-Eglais measure are given in Table 6. We can easily see that the results of the ESE-algorithm are better for almost all cases. It is likely that by running ESE for some more starting solutions, better or equally good designs can be found for all cases. The ESE algorithm thus outperforms the periodic designs for the Audze-Eglais measure.

When we compare the results with those found by Bates in Table 4, we see that the ESE-algorithm gives better or equally good results. This shows that the ESE-algorithm is quite successful in finding LHDs with a good Audze-Eglais value.

Table 4 Audze-Eglais values of designs found by Bates et al. (2004) vs. the ESE-algorithm

n	2 dim		3 dim		5 dim	
	PermGA	ESE	PermGA	ESE	PermGA	ESE
5	**1.2982**	**1.2982**	**0.7267**	**0.7267**		
10	**2.0662**	**2.0662**	1.0242	**1.0199**		
50					0.7270	**0.7195**
120	5.5174	**5.4941**	1.9613	**1.9328**	0.7930	**0.7840**

5 Conclusions

This paper discusses existing and new results in the field of maximin and Audze-Eglais Latin hypercube designs. Such designs play an important role in the area of computer simulation. The new results are obtained using two heuristics. The first heuristic is based on the observation that many optimal LHDs, and two-dimensional LHDs in particular, exhibit a periodic structure. By considering periodic and adapted periodic designs, approximate maximin LHDs for up to seven dimensions and for up to 300 design points are constructed. The second heuristic uses the ESE-algorithm of Jin et al. (2005) to find approximate maximin LHDs for up to ten dimensions. These new results are compared to each other and to existing results obtained with simulated annealing and permutation genetic algorithms. In most cases, the ESE-algorithm resulted in the best maximin and Audze-Eglais LHDs. However when the number of design points is large with respect to the dimension of the design, the periodic designs tend to work better. Appendix gives the squared ℓ^2-separation distances and Audze-Eglais values of the designs described in this paper. All corresponding designs can be downloaded from the website http://www.spacefillingdesigns.nl.

Acknowledgements The authors would like to thank the anonymous referees for their valuable comments.

Appendix: Tables of numerical results

Table 5 Squared ℓ^2-separation distance found using periodic designs (PD) vs. the ESE-algorithm (ESE)

n	2 dim		3 dim		4 dim		5 dim		6 dim		7 dim		8 dim	9 dim	10 dim
	ESE	Per	ESE	Per	ESE	Per	ESE	Per	ESE	Per	ESE	Per	ESE	ESE	ESE
2	**2**	**2**	**3**	**3**	**4**	**4**	**5**	**5**	**6**	**6**	**7**	**7**	**8**	**9**	**10**
3	**2**	**2**	**6**	3	**7**	4	**8**	5	**12**	6	**13**	7	**14**	**18**	**19**
4	**5**	**5**	**6**	**6**	**12**	**12**	**14**	11	**20**	15	**21**	16	**26**	**28**	**33**
5	**5**	**5**	**11**	6	**15**	12	**24**	11	**27**	15	**32**	16	**40**	**43**	**50**
6	**5**	**5**	**14**	**14**	**22**	16	**32**	23	**40**	28	**47**	29	**53**	**61**	**68**
7	**8**	**8**	**17**	14	**28**	16	**40**	23	**52**	28	**61**	31	**70**	**80**	**89**

Table 5 (*Continued*)

n	2 dim		3 dim		4 dim		5 dim		6 dim		7 dim		8 dim	9 dim	10 dim
	ESE	Per	ESE	Per	ESE	Per	ESE	Per	ESE	Per	ESE	Per	ESE	ESE	ESE
8	**8**	**8**	**21**	**21**	**42**	25	**50**	32	**63**	42	**79**	46	**90**	**101**	**114**
9	8	**10**	**22**	21	**42**	25	**61**	39	**75**	45	**92**	47	**112**	**126**	**142**
10	**10**	**10**	**27**	21	**47**	36	**82**	55	**91**	62	**109**	68	**131**	**154**	**171**
11	**10**	**10**	**30**	24	**55**	39	**80**	55	**108**	62	**129**	69	**152**	**178**	**206**
12	10	**13**	**36**	30	**63**	46	**91**	62	**136**	91	**152**	95	**177**	**204**	**235**
13	10	**13**	**41**	35	**70**	51	**103**	64	**138**	91	**178**	95	**205**	**235**	**268**
14	10	**17**	**42**	35	**77**	70	**114**	86	**154**	104	**215**	119	**236**	**268**	**305**
15	13	**17**	**45**	42	**87**	71	**129**	88	**171**	111	**220**	129	**273**	**309**	**347**
16	**17**	**17**	**50**	42	**93**	85	**151**	101	**190**	130	**241**	155	**317**	**352**	**393**
17	17	**18**	**53**	42	**99**	85	**158**	113	**208**	131	**266**	161	**332**	**396**	**442**
18	17	**18**	**56**	50	**108**	94	**170**	123	**231**	155	**291**	186	**361**	**451**	**496**
19	17	**18**	**59**	57	**119**	94	**184**	136	**256**	169	**323**	195	**390**	**469**	**554**
20	17	**18**	**65**	57	**130**	106	**206**	139	**279**	210	**349**	226	**425**	**506**	**625**
21	18	**20**	**68**	65	**145**	116	**223**	165	**302**	210	**380**	236	**463**	**548**	**650**
22	18	**25**	**72**	69	**150**	117	**235**	174	**325**	223	**418**	270	**501**	**595**	**691**
23	18	**26**	**75**	72	**159**	130	**250**	178	**348**	236	**448**	273	**542**	**640**	**747**
24	20	**26**	**81**	76	**170**	138	**266**	201	**374**	258	**481**	308	**585**	**690**	**800**
25	20	**26**	86	**91**	**178**	156	**285**	205	**400**	286	**520**	350	**626**	**739**	**857**
26	25	**26**	86	**91**	**188**	156	**302**	226	**426**	296	**548**	365	**664**	**791**	**910**
27	25	**26**	90	**91**	**198**	157	**310**	238	**447**	310	**585**	382	**712**	**840**	**976**
28	26	**29**	**94**	**94**	**210**	174	**331**	258	**479**	339	**620**	406	**766**	**898**	**1041**
29	26	**29**	**101**	94	**221**	174	**349**	269	**507**	346	**654**	417	**817**	**956**	**1100**
30	26	**29**	**105**	**105**	**233**	194	**367**	310	**531**	390	**691**	458	**849**	**1019**	**1173**
31	26	**32**	**110**	107	**244**	212	**405**	310	**563**	390	**728**	482	**900**	**1104**	**1241**
32	26	**32**	110	**114**	**253**	212	**413**	341	**587**	419	**778**	518	**966**	**1139**	**1318**
33	26	**34**	**117**	114	**264**	215	**426**	341	**622**	430	**814**	537	**1010**	**1201**	**1396**
34	26	**37**	125	**133**	**273**	230	**445**	358	**648**	470	**851**	561	**1072**	**1270**	**1478**
35	26	**37**	126	**133**	**286**	234	**467**	366	**683**	495	**914**	586	**1113**	**1326**	**1555**
36	26	**37**	131	**133**	**297**	250	**486**	400	**719**	518	**939**	636	**1181**	**1405**	**1647**
37	26	**37**	138	**152**	**309**	266	**520**	408	**744**	528	**976**	668	**1236**	**1477**	**1721**
38	26	**41**	142	**152**	**321**	283	**541**	415	**788**	561	**1028**	709	**1286**	**1534**	**1790**
39	26	**41**	146	**152**	**330**	283	**566**	439	**816**	561	**1084**	726	**1344**	**1609**	**1870**
40	26	**41**	152	**155**	**342**	291	**575**	492	**876**	632	**1122**	786	**1416**	**1675**	**1946**
41	26	**41**	158	**162**	**355**	293	**596**	492	**882**	632	**1156**	802	**1496**	**1765**	**2058**
42	29	**41**	161	**168**	**367**	319	**626**	496	**907**	670	**1209**	903	**1526**	**1843**	**2149**
43	29	**41**	**171**	168	**383**	323	**666**	520	**947**	670	**1256**	903	**1597**	**1905**	**2224**
44	29	**50**	179	**186**	**396**	331	**680**	548	**992**	696	**1336**	903	**1653**	**1994**	**2319**
45	37	**50**	182	**186**	**407**	347	**698**	565	**996**	737	**1366**	926	**1723**	**2079**	**2415**
46	37	**50**	186	**189**	**421**	366	**723**	592	**1064**	797	**1408**	985	**1794**	**2155**	**2507**
47	37	**50**	**189**	**189**	**438**	378	**754**	611	**1088**	797	**1459**	985	**1847**	**2244**	**2600**
48	37	**50**	**201**	189	**450**	413	**763**	632	**1119**	857	**1531**	1054	**1924**	**2336**	**2732**

Table 5 (*Continued*)

n	2 dim		3 dim		4 dim		5 dim		6 dim		7 dim		8 dim	9 dim	10 dim
	ESE	Per	ESE	Per	ESE	Per	ESE	Per	ESE	Per	ESE	Per	ESE	ESE	ESE
49	37	**50**	**203**	196	**464**	415	**803**	634	**1167**	893	**1592**	1074	**1989**	**2397**	**2828**
50	37	**52**	206	**213**	**478**	415	**830**	663	**1203**	893	**1639**	1113	**2041**	**2492**	**2893**
51	37	**52**	206	**213**	**490**	421	**850**	692	**1230**	917	**1662**	1161	**2132**	**2566**	**3006**
52	37	**58**	**217**	213	**504**	455	**883**	709	**1274**	1003	**1734**	1231	**2203**	**2686**	**3134**
53	37	**58**	**219**	216	**515**	455	**894**	716	**1340**	1003	**1808**	1241	**2234**	**2713**	**3261**
54	37	**58**	209	**233**	**534**	477	**932**	760	**1359**	1019	**1856**	1288	**2356**	**2805**	**3339**
55	40	**58**	230	**243**	**546**	483	**956**	760	**1421**	1082	**1896**	1325	**2429**	**2935**	**3452**
56	41	**58**	230	**243**	**558**	515	**982**	784	**1431**	1104	**2003**	1358	**2444**	**3021**	**3551**
57	41	**58**	249	**261**	**574**	515	**1007**	846	**1488**	1136	**2024**	1479	**2554**	**3119**	**3651**
58	41	**61**	245	**261**	**594**	539	**1035**	846	**1554**	1166	**2043**	1479	**2650**	**3187**	**3795**
59	41	**61**	254	**266**	**609**	544	**1063**	849	**1564**	1223	**2136**	1509	**2733**	**3297**	**3889**
60	41	**65**	261	**273**	**618**	568	**1094**	904	**1631**	1242	**2232**	1577	**2796**	**3420**	**4090**
61	41	**65**	266	**274**	**630**	620	**1128**	904	**1667**	1258	**2266**	1615	**2868**	**3525**	**4158**
62	41	**65**	269	**283**	**657**	620	**1150**	934	**1715**	1306	**2345**	1680	**2977**	**3636**	**4313**
63	50	**65**	281	**297**	**670**	620	**1178**	967	**1781**	1380	**2376**	1680	**3056**	**3690**	**4355**
64	50	**65**	278	**297**	**684**	625	**1206**	985	**1804**	1430	**2452**	1769	**3097**	**3820**	**4514**
65	50	**68**	290	**314**	**694**	630	**1216**	997	**1868**	1430	**2492**	1786	**3219**	**3932**	**4581**
66	50	**68**	299	**314**	**718**	666	**1261**	1050	**1874**	1476	**2543**	1857	**3279**	**4004**	**4769**
67	50	**74**	294	**314**	**735**	666	**1299**	1072	**1954**	1482	**2638**	1868	**3399**	**4081**	**4942**
68	50	**74**	306	**314**	**746**	685	**1330**	1087	**1983**	1538	**2693**	1940	**3453**	**4212**	**4995**
69	50	**74**	306	**324**	**765**	698	**1351**	1112	**2028**	1588	**2746**	1965	**3520**	**4317**	**5127**
70	50	**74**	314	**325**	**779**	716	**1378**	1150	**2094**	1633	**2838**	2130	**3588**	**4464**	**5276**
71	50	**74**	314	**325**	**793**	716	**1413**	1150	**2141**	1644	**2871**	2130	**3749**	**4548**	**5437**
72	50	**74**	314	**341**	**810**	750	**1430**	1203	**2136**	1768	**2960**	2177	**3810**	**4666**	**5556**
73	50	**74**	329	**350**	**834**	759	**1462**	1229	**2197**	1768	**3042**	2206	**3932**	**4776**	**5661**
74	50	**74**	341	**350**	**842**	767	**1512**	1229	**2291**	1774	**3120**	2244	**3941**	**4915**	**5817**
75	50	**80**	341	**350**	**867**	771	**1530**	1274	**2303**	1862	**3157**	2295	**4073**	**5006**	**5937**
76	50	**85**	341	**363**	**882**	813	**1569**	1300	**2387**	1935	**3218**	2375	**4178**	**5179**	**6111**
77	50	**85**	341	**363**	**894**	823	**1591**	1308	**2433**	1947	**3323**	2403	**4266**	**5222**	**6272**
78	50	**85**	371	**387**	**910**	844	**1621**	1382	**2479**	2014	**3387**	2505	**4390**	**5385**	**6384**
79	50	**85**	374	**387**	**927**	848	**1639**	1382	**2498**	2037	**3474**	2525	**4465**	**5535**	**6466**
80	50	**85**	374	**403**	**949**	873	**1691**	1395	**2554**	2037	**3550**	2590	**4565**	**5577**	**6653**
81	50	**85**	381	**406**	**963**	916	**1730**	1406	**2648**	2064	**3619**	2642	**4679**	**5748**	**6780**
82	50	**85**	374	**406**	**989**	938	**1742**	1475	**2680**	2141	**3669**	2753	**4719**	**5859**	**6935**
83	50	**90**	374	**417**	**1002**	940	**1762**	1501	**2696**	2141	**3723**	2767	**4848**	**5976**	**7094**
84	50	**90**	406	**426**	**1021**	967	**1818**	1534	**2790**	2229	**3870**	2838	**4920**	**6119**	**7256**
85	50	**90**	413	**426**	**1043**	967	**1866**	1552	**2819**	2232	**3919**	2874	**5032**	**6212**	**7357**
86	50	**97**	413	**428**	**1053**	967	**1882**	1573	**2875**	2375	**3958**	3103	**5164**	**6346**	**7532**
87	50	**97**	413	**428**	**1073**	976	**1934**	1598	**2913**	2375	**4095**	3103	**5225**	**6469**	**7639**
88	50	**97**	434	**437**	**1086**	1050	**1954**	1685	**2975**	2398	**4166**	3183	**5340**	**6660**	**7877**
89	50	**97**	426	**443**	**1102**	1050	**1990**	1690	**3067**	2400	**4176**	3183	**5450**	**6750**	**7950**

Table 5 (*Continued*)

n	2 dim		3 dim		4 dim		5 dim		6 dim		7 dim		8 dim	9 dim	10 dim
	ESE	Per	ESE	Per	ESE	Per	ESE	Per	ESE	Per	ESE	Per	ESE	ESE	ESE
90	58	**98**	446	**481**	**1134**	1060	**2027**	1710	**3104**	2516	**4308**	3190	**5576**	**6901**	**8128**
91	58	**98**	434	**481**	**1134**	1089	**2031**	1748	**3143**	2516	**4379**	3234	**5626**	**6950**	**8330**
92	58	**98**	446	**481**	**1149**	1089	**2100**	1805	**3216**	2599	**4428**	3277	**5758**	**7067**	**8442**
93	58	**100**	446	**481**	**1171**	1098	**2130**	1813	**3283**	2604	**4512**	3361	**5832**	**7342**	**8601**
94	58	**100**	470	**481**	**1199**	1124	**2169**	1881	**3348**	2747	**4581**	3474	**6007**	**7436**	**8774**
95	65	**100**	**482**	481	**1219**	1135	**2206**	1901	**3335**	2747	**4703**	3531	**6064**	**7469**	**8877**
96	65	**101**	486	**509**	1250	**1261**	**2227**	1965	**3451**	2769	**4808**	3639	**6222**	**7645**	**9146**
97	65	**101**	474	**515**	1258	**1261**	**2299**	1965	**3514**	2817	**4848**	3639	**6304**	**7781**	**9379**
98	65	**101**	485	**531**	**1283**	1261	**2299**	1965	**3560**	2850	**4936**	3690	**6376**	**7896**	**9381**
99	65	**101**	489	**531**	**1298**	1261	**2338**	2009	**3628**	2878	**4999**	3731	**6448**	**8023**	**9617**
100	65	**109**	494	**554**	**1305**	1261	**2401**	2053	**3648**	3000	**5040**	3903	**6617**	**8228**	**9835**
105			521	**563**	**1395**	1329									
110			566	**626**	**1510**	1414									
115			594	**650**	**1591**	1499									
120			629	**702**	**1708**	1603									
125			629	**713**	**1798**	1750									
130			693	**766**	**1906**	1872									
135			729	**780**	**1995**	1909									
140			758	**845**	**2103**	2089									
145			779	**894**	2185	**2225**									
150			825	**934**	**2310**	2278									
155			842	**986**	2365	**2367**									
160			854	**1002**	2486	**2548**									
165			904	**1041**	2582	**2648**									
170			914	**1121**	2659	**2869**									
175			965	**1132**	2771	**2902**									
180			1011	**1208**	2897	**3077**									
185			1026	**1224**	2970	**3267**									
190			1061	**1298**	3094	**3325**									
195			1086	**1350**	3210	**3492**									
200			1106	**1371**	3257	**3596**									
205			1166	**1425**	3273	**3708**									
210			1196	**1473**	3377	**3767**									
215			1229	**1538**	3476	**3983**									
220			1259	**1544**	3543	**4159**									
225			1293	**1611**	3661	**4292**									
230			1329	**1646**	3703	**4326**									
235			1305	**1706**	3815	**4532**									
240			1350	**1806**	3893	**5061**									
245			1397	**1891**	3986	**5061**									
250			1412	**1901**	3990	**5075**									

Table 5 (*Continued*)

n	2 dim		3 dim		4 dim		5 dim		6 dim		7 dim		8 dim	9 dim	10 dim
	ESE	Per	ESE	Per	ESE	Per	ESE	Per	ESE	Per	ESE	Per	ESE	ESE	ESE
255			1417	**1923**	4100	**5122**									
260			1445	**1971**	4164	**5236**									
265			1449	**2021**	4182	**5519**									
270			1464	**2144**	4361	**5656**									
275			1478	**2150**	4487	**5746**									
280			1493	**2184**	4388	**6023**									
285			1501	**2209**	4607	**6094**									
290			1476	**2269**	4722	**6380**									
295			1526	**2354**	4726	**6590**									
300			1542	**2409**	4898	**6604**									

Table 6 Audze-Eglais values found using periodic designs (PD) vs. the ESE-algorithm (ESE)

n	2 dim		3 dim		4 dim		5 dim		6 dim	7 dim	8 dim	9 dim	10 dim
	ESE	Per	ESE	Per	ESE	Per	ESE	Per	ESE	ESE	ESE	ESE	ESE
2	**0.500**	**0.500**	**0.333**	**0.333**	**0.250**	**0.250**	**0.200**	**0.200**	**0.167**	**0.143**	**0.125**	**0.111**	**0.100**
3	**0.900**	**0.900**	**0.611**	**0.611**	**0.386**	0.450	**0.321**	0.362	**0.250**	**0.230**	**0.193**	**0.200**	**0.151**
4	**1.000**	**1.000**	**0.642**	**0.642**	**0.454**	0.489	**0.367**	0.382	**0.300**	**0.260**	**0.225**	**0.201**	**0.180**
5	**1.298**	1.390	**0.727**	0.891	**0.509**	0.658	**0.401**	0.527	**0.336**	**0.287**	**0.250**	**0.222**	**0.200**
6	**1.521**	**1.521**	**0.794**	0.800	**0.561**	0.594	**0.431**	0.476	**0.358**	**0.307**	**0.268**	**0.238**	**0.215**
7	**1.598**	**1.598**	**0.867**	0.975	**0.599**	0.694	**0.464**	0.532	**0.376**	**0.322**	**0.282**	**0.250**	**0.225**
8	**1.804**	1.879	**0.921**	0.960	**0.619**	0.696	**0.488**	0.538	**0.398**	**0.334**	**0.292**	**0.260**	**0.234**
9	**1.935**	**1.935**	**0.971**	1.052	**0.660**	0.742	**0.504**	0.567	**0.414**	**0.349**	**0.301**	**0.267**	**0.240**
10	**2.066**	**2.066**	**1.020**	1.085	**0.686**	0.744	**0.515**	0.556	**0.425**	**0.360**	**0.311**	**0.273**	**0.246**
11	**2.196**	2.279	**1.069**	1.137	**0.709**	0.785	**0.536**	0.612	**0.434**	**0.369**	**0.319**	**0.281**	**0.250**
12	**2.273**	**2.273**	**1.095**	1.163	**0.724**	0.785	**0.551**	0.589	**0.441**	**0.375**	**0.326**	**0.287**	**0.256**
13	**2.401**	2.487	**1.128**	1.191	**0.746**	0.825	**0.563**	0.632	**0.453**	**0.381**	**0.331**	**0.292**	**0.261**
14	**2.476**	**2.476**	**1.167**	1.252	**0.762**	0.829	**0.575**	0.635	**0.462**	**0.385**	**0.335**	**0.296**	**0.265**
15	**2.578**	2.643	**1.194**	1.255	**0.775**	0.818	**0.583**	0.636	**0.470**	**0.393**	**0.339**	**0.299**	**0.268**
16	**2.666**	2.683	**1.221**	1.290	**0.791**	0.848	**0.589**	0.642	**0.477**	**0.398**	**0.341**	**0.302**	**0.271**
17	**2.721**	**2.721**	**1.246**	1.340	**0.805**	0.866	**0.600**	0.656	**0.483**	**0.404**	**0.347**	**0.305**	**0.273**
18	**2.819**	2.848	**1.271**	1.337	**0.816**	0.875	**0.609**	0.655	**0.488**	**0.408**	**0.350**	**0.307**	**0.275**
19	**2.890**	2.984	**1.292**	1.374	**0.827**	0.895	**0.615**	0.667	**0.492**	**0.413**	**0.354**	**0.310**	**0.277**
20	**2.959**	2.962	**1.318**	1.394	**0.835**	0.907	**0.620**	0.681	**0.496**	**0.416**	**0.358**	**0.313**	**0.278**
21	**3.025**	3.033	**1.339**	1.408	**0.847**	0.914	**0.625**	0.671	**0.501**	**0.419**	**0.361**	**0.316**	**0.281**
22	**3.070**	**3.070**	**1.357**	1.426	**0.856**	0.922	**0.632**	0.687	**0.505**	**0.422**	**0.363**	**0.318**	**0.283**
23	**3.138**	3.159	**1.377**	1.454	**0.868**	0.925	**0.638**	0.693	**0.510**	**0.425**	**0.366**	**0.321**	**0.285**
24	**3.197**	3.201	**1.396**	1.458	**0.875**	0.931	**0.644**	0.677	**0.513**	**0.427**	**0.368**	**0.323**	**0.287**
25	**3.254**	3.293	**1.412**	1.485	**0.884**	0.940	**0.648**	0.701	**0.516**	**0.430**	**0.370**	**0.324**	**0.289**

Table 6 (*Continued*)

n	2 dim		3 dim		4 dim		5 dim		6 dim	7 dim	8 dim	9 dim	10 dim
	ESE	Per	ESE	Per	ESE	Per	ESE	Per	ESE	ESE	ESE	ESE	ESE
26	**3.309**	3.332	**1.428**	1.480	**0.891**	0.947	**0.653**	0.707	**0.518**	**0.432**	**0.372**	**0.326**	**0.290**
27	**3.360**	3.383	**1.442**	1.499	**0.898**	0.957	**0.657**	0.708	**0.521**	**0.435**	**0.373**	**0.328**	**0.292**
28	**3.405**	3.420	**1.454**	1.503	**0.906**	0.961	**0.660**	0.712	**0.524**	**0.437**	**0.375**	**0.329**	**0.293**
29	**3.458**	3.539	**1.468**	1.543	**0.912**	0.978	**0.664**	0.716	**0.527**	**0.439**	**0.376**	**0.330**	**0.294**
30	**3.505**	3.515	**1.481**	1.528	**0.919**	0.974	**0.667**	0.716	**0.530**	**0.441**	**0.378**	**0.331**	**0.295**
31	**3.543**	3.550	**1.493**	1.563	**0.925**	0.976	**0.671**	0.719	**0.533**	**0.443**	**0.380**	**0.333**	**0.296**
32	**3.589**	3.623	**1.505**	1.562	**0.931**	0.996	**0.674**	0.729	**0.535**	**0.444**	**0.381**	**0.334**	**0.297**
33	**3.636**	3.642	**1.517**	1.588	**0.935**	0.990	**0.678**	0.732	**0.537**	**0.446**	**0.383**	**0.335**	**0.298**
34	**3.676**	3.713	**1.528**	1.565	**0.941**	1.005	**0.682**	0.735	**0.540**	**0.447**	**0.384**	**0.336**	**0.299**
35	**3.716**	3.786	**1.539**	1.601	**0.946**	1.003	**0.685**	0.731	**0.542**	**0.449**	**0.385**	**0.337**	**0.300**
36	**3.758**	3.774	**1.549**	1.600	**0.950**	1.005	**0.688**	0.734	**0.543**	**0.450**	**0.386**	**0.338**	**0.301**
37	**3.794**	3.819	**1.558**	1.599	**0.956**	1.019	**0.691**	0.736	**0.545**	**0.452**	**0.387**	**0.339**	**0.301**
38	**3.828**	**3.828**	**1.568**	1.623	**0.959**	1.023	**0.694**	0.746	**0.547**	**0.453**	**0.388**	**0.340**	**0.302**
39	**3.868**	3.879	**1.578**	1.646	**0.965**	1.025	**0.696**	0.742	**0.548**	**0.455**	**0.389**	**0.341**	**0.303**
40	**3.906**	3.918	**1.587**	1.636	**0.968**	1.019	**0.699**	0.742	**0.550**	**0.456**	**0.390**	**0.341**	**0.303**
41	**3.939**	4.009	**1.596**	1.639	**0.971**	1.033	**0.701**	0.751	**0.551**	**0.457**	**0.391**	**0.342**	**0.304**
42	**3.974**	3.974	**1.604**	1.658	**0.975**	1.031	**0.703**	0.742	**0.552**	**0.458**	**0.392**	**0.343**	**0.305**
43	**4.007**	4.045	**1.612**	1.675	**0.979**	1.042	**0.705**	0.752	**0.554**	**0.460**	**0.393**	**0.344**	**0.306**
44	**4.029**	**4.029**	**1.621**	1.670	**0.983**	1.040	**0.708**	0.754	**0.555**	**0.461**	**0.394**	**0.344**	**0.306**
45	**4.063**	4.074	**1.628**	1.678	**0.986**	1.044	**0.710**	0.752	**0.557**	**0.462**	**0.394**	**0.345**	**0.307**
46	**4.096**	4.115	**1.636**	1.693	**0.990**	1.044	**0.712**	0.753	**0.559**	**0.463**	**0.395**	**0.346**	**0.307**
47	**4.130**	4.179	**1.643**	1.695	**0.993**	1.055	**0.714**	0.761	**0.560**	**0.464**	**0.396**	**0.346**	**0.308**
48	**4.160**	4.206	**1.650**	1.699	**0.997**	1.052	**0.716**	0.759	**0.561**	**0.464**	**0.397**	**0.347**	**0.308**
49	**4.187**	**4.187**	**1.657**	1.711	**1.001**	1.059	**0.718**	0.762	**0.563**	**0.465**	**0.398**	**0.347**	**0.309**
50	**4.216**	4.254	**1.665**	1.713	**1.004**	1.058	**0.720**	0.765	**0.564**	**0.466**	**0.398**	**0.348**	**0.309**
51	**4.246**	4.280	**1.671**	1.729	**1.007**	1.063	**0.721**	0.768	**0.566**	**0.467**	**0.399**	**0.348**	**0.310**
52	**4.273**	4.277	**1.678**	1.730	**1.010**	1.062	**0.723**	0.765	**0.567**	**0.468**	**0.400**	**0.349**	**0.310**
53	**4.302**	4.343	**1.685**	1.734	**1.013**	1.072	**0.725**	0.771	**0.568**	**0.468**	**0.400**	**0.349**	**0.310**
54	**4.331**	4.341	**1.690**	1.739	**1.016**	1.070	**0.726**	0.769	**0.569**	**0.469**	**0.401**	**0.350**	**0.311**
55	**4.355**	4.413	**1.697**	1.755	**1.018**	1.073	**0.728**	0.773	**0.570**	**0.470**	**0.401**	**0.350**	**0.311**
56	**4.382**	4.404	**1.703**	1.756	**1.022**	1.071	**0.729**	0.772	**0.571**	**0.470**	**0.402**	**0.351**	**0.312**
57	**4.404**	4.427	**1.708**	1.760	**1.024**	1.079	**0.731**	0.776	**0.572**	**0.471**	**0.403**	**0.351**	**0.312**
58	**4.431**	4.437	**1.714**	1.763	**1.027**	1.076	**0.732**	0.776	**0.573**	**0.472**	**0.403**	**0.352**	**0.312**
59	**4.458**	4.498	**1.719**	1.777	**1.030**	1.087	**0.734**	0.780	**0.574**	**0.473**	**0.404**	**0.352**	**0.313**
60	**4.482**	4.490	**1.725**	1.772	**1.032**	1.079	**0.735**	0.777	**0.575**	**0.473**	**0.404**	**0.353**	**0.313**
61	**4.499**	4.530	**1.731**	1.778	**1.034**	1.087	**0.736**	0.776	**0.576**	**0.474**	**0.405**	**0.353**	**0.314**
62	**4.526**	4.576	**1.736**	1.786	**1.036**	1.087	**0.738**	0.781	**0.576**	**0.475**	**0.405**	**0.354**	**0.314**
63	**4.556**	4.576	**1.742**	1.789	**1.039**	1.094	**0.739**	0.783	**0.577**	**0.475**	**0.406**	**0.354**	**0.314**
64	**4.573**	4.590	**1.746**	1.794	**1.041**	1.087	**0.740**	0.784	**0.578**	**0.476**	**0.406**	**0.354**	**0.315**
65	**4.595**	4.599	**1.751**	1.802	**1.043**	1.095	**0.742**	0.786	**0.579**	**0.477**	**0.407**	**0.355**	**0.315**
66	**4.619**	4.635	**1.757**	1.804	**1.045**	1.093	**0.742**	0.785	**0.580**	**0.477**	**0.407**	**0.355**	**0.315**

Table 6 (*Continued*)

n	2 dim		3 dim		4 dim		5 dim		6 dim	7 dim	8 dim	9 dim	10 dim
	ESE	Per	ESE	Per	ESE	Per	ESE	Per	ESE	ESE	ESE	ESE	ESE
67	**4.636**	4.642	**1.761**	1.812	**1.047**	1.100	**0.744**	0.787	**0.581**	**0.478**	**0.407**	**0.355**	**0.315**
68	**4.661**	4.681	**1.766**	1.819	**1.049**	1.096	**0.745**	0.790	**0.581**	**0.478**	**0.407**	**0.356**	**0.316**
69	**4.683**	4.713	**1.771**	1.818	**1.052**	1.104	**0.746**	0.791	**0.582**	**0.479**	**0.408**	**0.356**	**0.316**
70	**4.703**	4.710	**1.775**	1.818	**1.053**	1.100	**0.747**	0.790	**0.583**	**0.480**	**0.408**	**0.356**	**0.316**
71	**4.727**	4.742	**1.780**	1.831	**1.055**	1.108	**0.747**	0.795	**0.584**	**0.480**	**0.409**	**0.357**	**0.317**
72	**4.743**	4.746	**1.784**	1.829	**1.057**	1.103	**0.749**	0.791	**0.584**	**0.481**	**0.409**	**0.357**	**0.317**
73	**4.763**	4.781	**1.789**	1.836	**1.059**	1.106	**0.749**	0.794	**0.585**	**0.481**	**0.409**	**0.357**	**0.317**
74	**4.781**	4.820	**1.793**	1.842	**1.061**	1.112	**0.751**	0.795	**0.586**	**0.482**	**0.410**	**0.358**	**0.317**
75	**4.803**	4.817	**1.796**	1.847	**1.063**	1.112	**0.752**	0.797	**0.586**	**0.482**	**0.410**	**0.358**	**0.318**
76	**4.823**	4.828	**1.801**	1.850	**1.064**	1.110	**0.753**	0.796	**0.587**	**0.483**	**0.410**	**0.358**	**0.318**
77	**4.838**	4.853	**1.805**	1.851	**1.066**	1.112	**0.755**	0.799	**0.587**	**0.483**	**0.411**	**0.359**	**0.318**
78	**4.863**	4.883	**1.809**	1.847	**1.068**	1.114	**0.755**	0.795	**0.588**	**0.484**	**0.411**	**0.359**	**0.318**
79	**4.882**	4.934	**1.812**	1.863	**1.070**	1.119	**0.756**	0.801	**0.589**	**0.484**	**0.411**	**0.359**	**0.318**
80	**4.895**	4.922	**1.816**	1.864	**1.071**	1.117	**0.757**	0.799	**0.589**	**0.484**	**0.412**	**0.359**	**0.319**
81	**4.920**	4.942	**1.820**	1.869	**1.072**	1.120	**0.758**	0.802	**0.590**	**0.485**	**0.412**	**0.360**	**0.319**
82	**4.936**	4.944	**1.824**	1.862	**1.074**	1.120	**0.759**	0.801	**0.590**	**0.485**	**0.413**	**0.360**	**0.319**
83	**4.949**	4.949	**1.827**	1.879	**1.076**	1.126	**0.760**	0.805	**0.591**	**0.486**	**0.413**	**0.360**	**0.319**
84	**4.968**	4.992	**1.831**	1.876	**1.077**	1.122	**0.761**	0.802	**0.591**	**0.486**	**0.413**	**0.360**	**0.320**
85	**4.985**	5.014	**1.834**	1.879	**1.079**	1.124	**0.761**	0.804	**0.592**	**0.486**	**0.414**	**0.360**	**0.320**
86	**5.003**	5.014	**1.838**	1.882	**1.081**	1.125	**0.762**	0.804	**0.592**	**0.487**	**0.414**	**0.361**	**0.320**
87	**5.019**	5.060	**1.842**	1.891	**1.082**	1.130	**0.763**	0.808	**0.593**	**0.487**	**0.414**	**0.361**	**0.320**
88	**5.034**	5.047	**1.845**	1.885	**1.083**	1.130	**0.764**	0.805	**0.594**	**0.487**	**0.414**	**0.361**	**0.320**
89	**5.056**	5.096	**1.848**	1.895	**1.085**	1.133	**0.765**	0.810	**0.594**	**0.488**	**0.415**	**0.361**	**0.321**
90	5.070	**5.063**	**1.852**	1.885	**1.086**	1.131	**0.766**	0.807	**0.594**	**0.488**	**0.415**	**0.361**	**0.321**
91	**5.086**	5.113	**1.854**	1.890	**1.088**	1.134	**0.766**	0.809	**0.595**	**0.489**	**0.415**	**0.362**	**0.321**
92	**5.104**	5.114	**1.858**	1.902	**1.089**	1.135	**0.767**	0.809	**0.595**	**0.489**	**0.416**	**0.362**	**0.321**
93	**5.119**	5.122	**1.861**	1.903	**1.090**	1.136	**0.768**	0.810	**0.596**	**0.489**	**0.416**	**0.362**	**0.321**
94	**5.130**	5.143	**1.864**	1.900	**1.092**	1.138	**0.769**	0.810	**0.596**	**0.490**	**0.416**	**0.362**	**0.321**
95	**5.151**	5.177	**1.867**	1.909	**1.093**	1.138	**0.769**	0.813	**0.597**	**0.490**	**0.416**	**0.362**	**0.322**
96	**5.163**	5.183	**1.870**	1.910	**1.094**	1.139	**0.770**	0.811	**0.597**	**0.490**	**0.417**	**0.363**	**0.322**
97	**5.177**	5.179	**1.872**	1.915	**1.096**	1.138	**0.771**	0.814	**0.598**	**0.490**	**0.417**	**0.363**	**0.322**
98	**5.198**	5.223	**1.876**	1.915	**1.097**	1.142	**0.771**	0.812	**0.598**	**0.491**	**0.417**	**0.363**	**0.322**
99	**5.211**	5.244	**1.879**	1.923	**1.098**	1.143	**0.772**	0.815	**0.599**	**0.491**	**0.417**	**0.363**	**0.322**
100	5.223	**5.221**	**1.882**	1.921	**1.099**	1.143	**0.773**	0.812	**0.599**	**0.491**	**0.418**	**0.363**	**0.322**

References

Alam FM, McNaught KR, Ringrose TJ (2004) A comparison of experimental designs in the development of a neural network simulation metamodel. Simul Model Pract Theory 12(7–8):559–578

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

Baer D (1992) Punktverteilungen in Würfeln beliebiger Dimension bezüglich der Maximum-norm. Wissenschaft. Z. Pädagog. Hochschule Erfurt/Mühlhausen, Mathematisch-Naturwissenschaftliche Reihe 28:87–92

Barthelemy JFM, Haftka RT (1993) Approximation concepts for optimum structural design—a review. Struct Multidiscip Optim 5(3):129–144

Bates RA, Buck RJ, Riccomagno E, Wynn HP (1996) Experimental design and observation for large systems. J R Stat Soc B 58:77–94

Bates SJ, Sienz J, Toropov VV (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, pp 1–7

Bulik M, Liefvendahl M, Stocki R, Wauquiez C (2004) Stochastic simulation for crashworthiness. Adv Eng Softw 35(12):791–803

Crary SB (2002) Design of computer experiments for metamodel generation. Analog Integr Circuits Signal Process 32(1):7–16

Crary SB (2008) WebDOE™. http://www.webdoe.cc. January 2008

Crary SB, Cousseau P, Armstrong D, Woodcock DM, Mok EH, Dubochet O, Lerch P, Renaud P (2000) Optimal design of computer experiments for metamodel generation using I-OPT™. Comput Model Eng Sci 1(1):127–139

van Dam ER (2008) Two-dimensional minimax Latin hypercube designs. Discrete Appl Math 156(18):3483–3493

van Dam ER, Husslage BGM, den Hertog D, Melissen JBM (2007) Maximin Latin hypercube designs in two dimensions. Oper Res 55(1):158–169

van Dam ER, Rennen G, Husslage BGM (2009) Bounds for maximin Latin hypercube designs. Oper Res 57:595–608

Dimnaku A, Kincaid R, Trosset MW (2005) Approximate solutions of continuous dispersion problems. Ann Oper Res 136(1):65–80

Driessen LT, Stehouwer HP, Wijker JJ (2002) Structural mass optimization of the engine frame of the Ariane 5 ESC-B. In: Proceedings of the European conference on spacecraft, structures, materials & mechanical testing, Toulouse, France, pp 1–9

Erkut E (1990) The discrete p-dispersion problem. Eur J Oper Res 46(1):48–60

Fejes Tóth L (1971) Punktverteilungen in einem Quadrat. Studia Sci Math Hung 6:439–442

Florian A (1989) Verteilung von Punkten in einem Quadrat. Sitzungsberichte, Abteilung II, Österreichische Akademie der Wissenschaften, Mathematisch-Naturwissenschaftliche Klasse 198:27–44

Forrester AIJ, Keane AJ, Bressloff NW (2006) Design and analysis of "noisy" computer experiments. AIAA J 44(10):2331–2339

Giunta AA, Wojtkiewicz SF, Eldred MS (2003) Overview of modern design of experiments methods for computational simulations. In: AIAA 2003, vol 649, pp 1–17

den Hertog D, Stehouwer HP (2002) Optimizing color picture tubes by high-cost nonlinear programming. Eur J Oper Res 140(2):197–211

Hino R, Yoshida F, Toropov VV (2006) Optimum blank design for sheet metal forming based on the interaction of high-and low-fidelity FE models. Arch Appl Mech 75(10):679–691

Husslage BGM (2006) Maximin designs for computer experiments. PhD thesis, CentER for Economic Research, Tilburg University, Tilburg, The Netherlands

Husslage BGM, Rennen G, van Dam ER, den Hertog D (2006) Space-filling Latin hypercube designs for computer experiments. CentER Discussion Paper 2006-18, pp 1–11. Tilburg University, Tilburg, The Netherlands

Jin R, Chen W, Sudjianto A (2005) An efficient algorithm for constructing optimal design of computer experiments. J Stat Plan Inference 134(1):268–287

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

Kirchner K, Wengerodt G (1987) Die dichteste Packung von 36 Kreisen in einem Quadrat. Beitrage Algebra Geom 25:147–159

Koehler JR, Owen AB (1996) Computer experiments. In: Ghosh S, Rao CR (eds) Design and analysis of experiments. Handbook of Statistics, vol 13. North-Holland, Amsterdam, pp 261–308

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

Locatelli M, Raber U (2002) Packing equal circles in a square: a deterministic global optimization approach. Discrete Appl Math 122(1–3):139–166

Markót MC, Csendes T (2005) A new verified optimization technique for the “packing circles in a unit square” problems. SIAM J Control Optim 16(1):193–219
McKay MD, Beckman RJ, Conover WJ (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21(2):239–245
Melissen JBM (1997) Packing and covering with circles. PhD thesis, Utrecht University, Utrecht, The Netherlands
Morris MD, Mitchell TJ (1995) Exploratory designs for computer experiments. J Stat Plan Inference 43:381–402
Nurmela KJ, Östergård PRJ (1999) More optimal packings of equal circles in a square. Discrete Comput Geom 22:439–547
Palmer K, Tsui KL (2001) A minimum bias Latin hypercube design. IIE Trans 33(9):793–808
Peikert R, Würtz D, Monagan M, den Groot C (1991) Packing circles in a sphere: a review and new results. In: Proceedings of the 15th IFIP conference on system modeling and optimization. Springer lecture notes in control and information sciences, vol 180, pp 111–124
Rennen G, Husslage BGM, van Dam ER, den Hertog D (2010) Nested maximin Latin hypercube designs. Struct Multidiscip Optim 46(2):287–306
Rikards R, Auzins J (2004) Response surface method for solution of structural identification problems. Inverse Probl Eng 12(1):59–70
Rikards R, Chate A, Gailis G (2001) Identification of elastic properties of laminates based on experiment design. Int J Solids Struct 38(30–31):5097–5115
Santner TJ, Williams BJ, Notz WI (2003) The design and analysis of computer experiments. Springer Series in Statistics. Springer, New York
Simpson TW, Booker AJ, Ghosh D, Giunta AA, Koch PN, Yang R-J (2004) Approximation methods in multidisciplinary analysis and optimization: a panel discussion. Struct Multidiscip Optim 27(5):302–313
Simpson TW, Peplinski J, Koch PN, Allen JK (2001) Metamodels for computer-based engineering design: survey and recommendations. Eng Comput 17:129–150
Sobieszczanski-Sobieski J, Haftka RT (1997) Multidisciplinary aerospace design optimization: survey of recent developments. Struct Multidiscip Optim 14(1):1–23
Specht E (2008) Packomania. http://www.packomania.com. January 2008
Stinstra ED, den Hertog D, Stehouwer HP, Vestjens A (2003) Constrained maximin designs for computer experiments. Technometrics 45(4):340–346
Stocki R (2005) A method to improve design reliability using optimal Latin hypercube sampling. Comput Assist Mech Eng Sci 12(4):393–412
Trosset MW (1999) Approximate maximin distance designs. In: Proceedings of the section on physical and engineering sciences, Alexandria, VA, USA, pp 223–227
Ye KQ, Li W, Sudjianto A (2000) Algorithmic construction of optimal symmetric Latin hypercube designs. J Stat Plan Inference 90(1):145–159