CentER

Discussion Paper

No. 2006–18

SPACE-FILLING LATIN HYPERCUBE DESIGNS FOR COMPUTER EXPERIMENTS

By Bart Husslage, Gijs Rennen, Edwin R. van Dam, Dick den Hertog

March 2006

ISSN 0924-7815

Tilburg University

Space-filling Latin hypercube designs for computer experiments

Bart Husslage* • Gijs Rennen • Edwin R. van Dam† • Dick den Hertog

Department of Econometrics and Operations Research, Tilburg University,
P.O. Box 90153, 5000 LE Tilburg, The Netherlands
B.G.M.Husslage@uvt.nl • G.Rennen@uvt.nl • Edwin.vanDam@uvt.nl • D.denHertog@uvt.nl

Abstract

In the area of computer simulation Latin hypercube designs play an important role. In this paper the class of maximin Latin hypercube designs is considered. Up to now only several two-dimensional designs and designs for some small number of points are known for this class. Using periodic designs and simulated annealing we extend the known results and construct approximate maximin Latin hypercube designs for up to ten dimensions and for up to 100 design points. All these designs can be downloaded from the website `http://www.spacefillingdesigns.nl`.

Keywords: Computer experiment, Latin hypercube design, non-collapsing, packing problem, simulated annealing, space-filling.
JEL Classification: C90.

1 Introduction

A k-dimensional Latin hypercube design (LHD) of n points, is a set of n points $x_i = (x_{i1}, x_{i2}, \ldots, x_{ik}) \in \{0, \ldots, n-1\}^k$ such that for each dimension j all x_{ij} are distinct. 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. Such maximin LHDs are very useful as designs for computer experiments. In this paper, we concentrate on the ℓ^2-distance measure since this measure is often the first choice in practice. We construct approximate maximin LHDs for up to ten dimensions and for up to 100 design points by using periodic designs and simulated annealing. All these designs can be downloaded from the website `http://www.spacefillingdesigns.nl`. As far as we know this is the first catalogue of maximin LHDs, although there are several catalogues for classical design of experiments, see e.g. the WebDOE$^{\text{TM}}$ website of Crary (2001).

Our main motivation for investigating this subject is that maximin Latin hypercube designs are extremely 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).

*The research of B.G.M. Husslage is funded 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.

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) and Morris and Mitchell (1995)). 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. 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. 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 maximin LHDs. For example, Morris and Mitchell (1995) use 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. (2006) derive 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 are obtained by using a branch-and-bound algorithm and constructing (adapted) periodic designs. Jin et al. (2005) describe an enhanced stochastic evolutionary algorithm for finding approximate maximin LHDs. They also apply their method to other space-filling criteria. Ye et al. (2000) propose an exchange algorithm for finding approximate maximin symmetric LHDs. The symmetry property is used as a compromise between computing effort and design optimality.

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 (2005). 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). Mladenovic et al. (2005) describe a method to find the densest packing of equal circles in a unit circle.

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.

There are several other measures proposed in the literature besides maximin, e.g. maximum entropy, minimax, IMSE, Audze-Eglais, and discrepancy. For a good overview, we refer to Koehler and Owen (1996). In statistical environments Latin hypercube sampling 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, Latin hypercube sampling, orthogonal array sampling, and Hammersley sequence sampling. However, for deterministic computer experiments it is better to use one of the measures mentioned above to get a better coverage of the design space. Bates et al. (1996) obtain 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) describe an enhanced stochastic evolutionary algorithm for finding maximum entropy and uniform designs. Van Dam (2005) derives interesting results for two-dimensional minimax LHDs. Bates et al. (2004) propose a permutation genetic algorithm to find optimal Audze-Eglais LHDs. Crary et al. (2000) developed I-OPT$^{\text{TM}}$ to generate LHDs with minimal IMSE. They found that IMSE-optimal designs can have proximate design points, which they call "twin points"; see also Crary (2002).

In literature different designs for computer experiments have been compared and the overall conclusion tends to be that the maximum entropy and distance-based criteria often perform best; see e.g. Simpson et al. (2001), Santner et al. (2003), and Bursztyn and Steinberg (2006).

This paper is organized as follows. Section 2 describes how periodic designs can be used to obtain good approximate maximin LHDs. A simulated annealing algorithm to construct such approximate maximin LHDs is discussed in Section 3. Computational results for up to ten dimensions and for up to 100 design points, as well as a comparison of both methods, are provided in Section 4. Finally, Section 5 contains conclusions.

2 Periodic designs

Van Dam et al. (2006) show 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 $y_1, \ldots, y_k$, with every y_i a permutation of the set $\{0, \ldots, n-1\}$. As in the two-dimensional case, a design is constructed by fixing the first dimension, without loss of generality, to the sequence $y_1 = (0, \ldots, n-1)$ and assigning (adapted) periodic sequences to all other dimensions. Two types of periodic sequences are considered. The first one is the sequence $(v_0, \ldots, v_{n-1})$, where

$$v_i = (s + ip) \bmod (n+1) - 1, \text{ for } i = 0, \ldots, n-1.$$

Here, s is the starting point of the sequence and p its period, which is chosen such that $\gcd(n+1, p) = 1$, 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 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 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

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

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 and used throughout the whole process. 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 designs are based on a shift of $1 - p$, -1, or 1 (as was the case for two dimensions; see Van Dam et al. (2006)). Furthermore, most 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 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	–	–	$2 \leq n \leq 100$

Table 1: Different classes of periodic sequences are checked for each dimension.

As an example, consider a three-dimensional adapted periodic LHD of 22 points. A best parameter setting is found to be $(p_2, q_2, s_2, m_2) = (8, -7, 7, 22)$ and $(p_3, q_3, s_3, m_3) = (3, 0, 3, 23)$ and, hence, the corresponding LHD, with separation distance 69, is defined by the sequences

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

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. $(0, 8, \ldots, 14)$, is formed by shifting the first part of y_2, i.e. $(7, 15, \ldots, 21)$, by -7. The periods and shift are clearly visible in the two-dimensional projection of the LHD in Figure 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 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; see Table 3.

3 Simulated annealing

Another heuristic method that can be used to approximate ℓ^2-maximin Latin hypercube designs is simulated annealing; see Aarts and Lenstra (1997). The general simulated annealing algorithm which we use is described in Algorithm 1. In this algorithm, we still need to specify the acceptance probability function, the annealing schedule, the terminating condition, and the neighborhood. All these parameters of the algorithm influence its performance. In this paper, we focus our attention on the choice of the neighborhood and the terminating condition. The chosen acceptance probability function is the commonly used classic formula by Kirkpatrick et al. (1983):

$$P\left(E_{\text{current}}, E_{\text{neighbor}}, T\right) = \exp\left(\frac{E_{\text{current}} - E_{\text{neighbor}}}{T}\right),$$

where E_{current} and E_{neighbor} are the separation distances of the current LHD and the neighbor LHD, respectively, and T is the temperature.

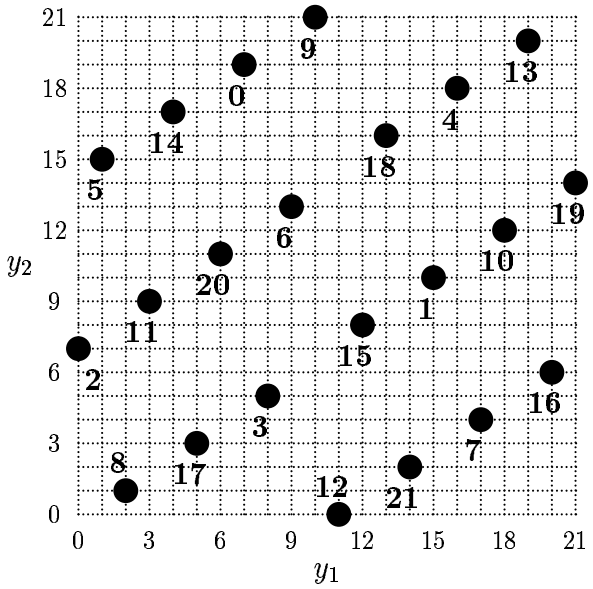

Figure 1: Two-dimensional projection of the three-dimensional LHD (y_1, y_2, y_3) of 22 points.

The annealing schedule starts with an initial temperature of 5. Each iteration the temperature is decreased by 0.1 percent, as long as the temperature is above 0.5. Furthermore, every 1,000 iterations the algorithm checks the number of improvements on the best solution found so far. If there were no improvements during the last 1,000 iterations, the temperature is reset by multiplying it by 2.7, which is approximately $0.999^{-1,000}$.

We have tried four different terminating conditions. The first two conditions terminate the algorithm after a fixed number of 25,000 and 50,000 iterations, respectively. The third and fourth condition let the number of iterations depend on the results of the algorithm in the following way. Every 1,000 iterations, it is checked whether the best design has improved. If during five subsequent checks, i.e. during the last 5,000 iterations, no improvement is made, the algorithm terminates. In order to avoid running times from becoming too large, the number of iterations is limited to 125,000 and 250,000 in the third and fourth condition, respectively.

For the simulated annealing algorithm, we have determined four different neighborhoods. In all four neighborhoods the main idea is to change two points of the current LHD by exchanging one or more

Algorithm 1 General simulated annealing algorithm for approximating ℓ^2-maximin LHDs

```
Randomly select an initial LHD and calculate its separation distance
Best LHD = initial LHD
REPEAT
  Create neighbor LHD of the current LHD
  Calculate separation distance of the neighbor LHD
  IF separation distance of neighbor LHD >= separation distance of current LHD
    Current LHD = neighbor LHD
    IF separation distance of current LHD >= separation distance of best LHD
      Best LHD = current LHD
    END
  ELSE with probability depending on temperature and difference in
       separation distance
    Current LHD = neighbor LHD
  END
  Update annealing temperature
UNTIL terminating condition is met
```

coordinate values. In three of the four neighborhoods, one point is required to be a critical point. A critical point is a point which is at separation distance to one of the other points.

In the first neighborhood, one point j_1 is selected randomly from all critical points and the other point j_2 randomly from all remaining points. This implies that the second point can either be a critical or non-critical point. Once the points are selected, the number of coordinates to change is randomly selected. Due to symmetry, at most $\lfloor \frac{k}{2} \rfloor$ coordinates are changed. Subsequently, the coordinates to change are randomly selected. The values of the two points in these coordinates are then exchanged, which results in a new LHD.

As an example, consider the four-dimensional LHD of 10 points defined by the sequences

$$\begin{array}{rcl} y_1 & = & (5,\ 6,\ 9,\ 3,\ 1,\ 4,\ 2,\ 8,\ 0,\ 7), \\ y_2 & = & (4,\ 5,\ 8,\ 6,\ 0,\ 2,\ 9,\ 7,\ 3,\ 1), \\ y_3 & = & (0,\ 4,\ 6,\ 1,\ 9,\ 7,\ 3,\ 5,\ 2,\ 8), \\ y_4 & = & (2,\ 3,\ 6,\ 5,\ 4,\ 9,\ 0,\ 7,\ 8,\ 1). \end{array}$$

The critical points of this design are points 3 and 8, i.e. $(9, 8, 6, 6)$ and $(8, 7, 5, 7)$. If the critical point $j_1 = 8$, the random point $j_2 = 4$, and the coordinates 2 and 3 are selected, the following neighbor is obtained:

$$\begin{array}{rcl} y_1 & = & (5,\ 6,\ 9,\ \ 3,\ \ 1,\ 4,\ 2,\ \ 8,\ \ 0,\ 7), \\ y_2 & = & (4,\ 5,\ 8,\ \boxed{7},\ 0,\ 2,\ 9,\ \boxed{6},\ 3,\ 1), \\ y_3 & = & (0,\ 4,\ 6,\ \boxed{5},\ 9,\ 7,\ 3,\ \boxed{1},\ 2,\ 8), \\ y_4 & = & (2,\ 3,\ 6,\ \ 5,\ \ 4,\ 9,\ 0,\ \ 7,\ \ 8,\ 1). \end{array}$$

The second neighborhood is very similar to the first. The only difference is that always one coordinate is selected instead of a random number of coordinates. Note that for $k = 3$ both neighborhoods are the same.

In the third neighborhood, also one coordinate is changed, however, now the coordinate is not randomly selected. Instead, all coordinates are tried and the one which results in the neighbor with the largest separation distance is selected. If more coordinates result in the same separation distance, the one with the lowest index is selected.

The fourth neighborhood is again very similar to the second neighborhood. The difference is that the first point is randomly selected from all points, instead of only the critical points.

Although the described approach appears to be quite similar to simulated annealing algorithms for finding good LHDs used by other authors, it is different in the following ways. Firstly, our approach does not impose a certain additional structure on the LHD, like, for instance, symmetry; see e.g. Ye et al. (2000). Secondly, the maximin distance criterion is used as the objective function. This in contrast to the approach of, for example, Morris and Mitchell (1995), who minimize a surrogate measure. The reason for using a surrogate measure is to minimize the number of critical points. The main disadvantage of this measure is, however, that it contains an extra parameter, which needs to be set for every value of k and n. An inaccurate setting of this parameter could lead to the situation where designs with a larger maximin distance have a larger value for the surrogate measure. On the other hand, a disadvantage of using the maximin distance criterion is that many designs may have the same objective value. However, we have reduced this problem by using neighborhoods that use critical points and by accepting equally good designs. By using critical points, we also implicitly try to reduce the number of critical points, without the need to introduce a surrogate measure.

4 Computational results

Periodic and adapted periodic designs have been constructed for up to seven dimensions and for up to 100 design points, using the different classes depicted in Table 1. Using simulated annealing, approximate maximin Latin hypercube designs have also been obtained for dimensions 8 to 10. All computations have been performed on PCs with a 800-MHz Pentium III processor. Table 2 shows the total CPU-times needed to construct approximate maximin Latin hypercube designs, for up to 100 points, for each dimension.

Although our heuristics only consider a subset of all possible Latin hypercube designs it can be seen from the table that still a considerable amount of time is needed to find good LHDs in higher dimensions and for a large number of points. Fortunately, however, these computation times are a one-time cost, i.e.

once a good design has been found, and its coordinates saved, the design can be used over and over again in various applications, without incurring the computational costs again.

dimension	3	4	5	6	7	8	9	10
CPU-time (hrs) PD	145	61	267	108	232	–	–	–
CPU-time (hrs) SA	500	181	152	520	246	460	470	470

Table 2: Total CPU-times needed to construct approximate maximin LHDs, up to 100 points, using periodic designs (PD) and simulated annealing (SA).

Table 3 shows the squared ℓ^2-maximin distances that were obtained by applying both heuristics. From this table it can be seen that (adapted) periodic designs work particularly well for larger values of n. For dimension 3 to 5 a break-even point, i.e. a point (or, better, an interval) where the preference shifts from the designs found by simulated annealing to (adapted) periodic designs, is clearly visible in the table. 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 $k \geq 6$, will occur for larger values of n, i.e. $n > 100$. This behavior could be explained by the "border effect", i.e. the irregularity of designs that is caused by the borders of the design space. Clearly, the number of "borders" of the k-dimensional box region increases exponentially, with respect to k. However, due to the Latin hypercube structure the number of design points that are located on or near these borders is limited. This, in turn, leads to very irregular optimal Latin hypercube designs when the number of design points is small with respect to the number of borders (which again depends on k). Hence, the nice, periodic structure that is sought for by our periodic heuristic only works well when the number of design points is relatively large, when compared to the dimension. Van Dam et al. (2006) already show the presence of this particular behavior in two-dimensional maximin Latin hypercube designs, i.e. the optimal designs found can all be represented by periodic designs. The results of Table 3 suggest that this behavior also occurs in higher dimensions. Simulated annealing, however, does not depend on an underlying structure and can therefore often find better designs, especially for smaller values of n. Since all six- and seven-dimensional (adapted) periodic designs, of 3 to 100 points, are dominated by the designs found by simulated annealing, maximin distances of the former are only computed for up to seven dimensions. Concerning the different neighborhoods for the simulated annealing algorithm (see Section 3), it turned out that the second neighborhood yields, in general, the best results. For the terminating conditions, the first two conditions, generally speaking, result in the best LHDs for $n \leq 50$, whereas the third and fourth condition are better for larger values of n.

Our heuristics are able to generate all best-known maximin Latin hypercube designs (see Morris and Mitchell (1995)), except for the cases $k = 6, n = 12$ and $k = 7, n = 14$, for which slightly worse designs are obtained. For the case $k = 3, n = 11$, however, we obtained an improved (and optimal) design. Furthermore, using a branch-and-bound algorithm, the three-dimensional designs of up to 13 points have been verified to be optimal.

5 Conclusions

This paper discusses two heuristics to obtain approximate maximin Latin hypercube designs. Such designs play an important role in the area of computer simulation. 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 100 design points are constructed. The second heuristic uses simulated annealing to find approximate maximin LHDs for up to ten dimensions. Although simulating annealing algorithms have been used before to deal with this type of problem, our adapted neighborhood structure, which is based on critical points, and the use of a different objective function, turned out to work particularly well. Combining both heuristics resulted in many new approximate maximin Latin hypercube designs. The periodic heuristic tends to work well when the number of design points is large, with respect to the dimension of the design, whereas the simulated annealing algorithm performs better for smaller values. The obtained squared ℓ^2-maximin distances are provided in Table 3. All corresponding approximate ℓ^2-maximin Latin hypercube designs can be downloaded from the website `http://www.spacefillingdesigns.nl`.

n	3 dim		4 dim		5 dim		6 dim		7 dim		8 dim	9 dim	10 dim
	PD	SA	PD	SA	PD	SA	PD	SA	PD	SA	SA	SA	SA
2	**3**	**3**	**4**	**4**	**5**	**5**	**6**	**6**	**7**	**7**	**8**	**9**	**10**
3	3	**6**	4	**7**	5	**8**	6	**12**	7	**13**	**14**	**18**	**19**
4	**6**	**6**	**12**	**12**	11	**14**	15	**20**	16	**21**	**26**	**28**	**33**
5	6	**11**	12	**15**	11	**24**	15	**27**	16	**32**	**40**	**43**	**50**
6	**14**	**14**	16	**22**	23	**32**	28	**40**	29	**47**	**54**	**61**	**68**
7	14	**17**	16	**28**	23	**40**	28	**52**	31	**61**	**70**	**80**	**89**
8	**21**	**21**	25	**42**	32	**50**	42	**66**	46	**79**	**91**	**101**	**114**
9	21	**22**	25	**42**	39	**61**	45	**76**	47	**93**	**112**	**126**	**141**
10	21	**27**	36	**50**	55	**82**	62	**91**	68	**110**	**130**	**154**	**172**
11	24	**30**	39	**55**	55	**80**	62	**108**	69	**128**	**152**	**178**	**206**
12	30	**36**	46	**63**	62	**91**	91	**136**	95	**150**	**176**	**204**	**235**
13	35	**41**	51	**68**	64	**101**	91	**136**	95	**174**	**202**	**232**	**267**
14	35	**42**	70	**75**	86	**112**	104	**152**	119	**204**	**228**	**265**	**298**
15	42	**48**	71	**83**	88	**124**	111	**167**	129	**211**	**257**	**296**	**337**
16	42	**50**	85	**90**	101	**136**	130	**186**	155	**238**	**286**	**330**	**378**
17	42	**53**	85	**97**	113	**150**	131	**203**	161	**256**	**312**	**367**	**415**
18	50	**56**	94	**103**	123	**162**	155	**223**	186	**281**	**344**	**398**	**458**
19	57	**59**	94	**113**	136	**174**	169	**241**	195	**305**	**370**	**438**	**498**
20	57	**62**	106	**123**	139	**184**	210	**260**	226	**332**	**403**	**472**	**542**
21	65	**66**	116	**127**	165	**201**	210	**283**	236	**361**	**438**	**517**	**592**
22	**69**	**69**	117	**137**	174	**215**	223	**304**	270	**384**	**467**	**555**	**643**
23	72	**74**	130	**146**	178	**224**	236	**324**	273	**410**	**501**	**596**	**685**
24	76	**78**	138	**154**	201	**242**	258	**343**	308	**444**	**538**	**639**	**739**
25	**91**	81	156	**162**	205	**255**	286	**368**	350	**467**	**583**	**688**	**792**
26	**91**	86	156	**171**	226	**269**	296	**387**	365	**499**	**612**	**726**	**854**
27	**91**	90	157	**178**	238	**287**	310	**410**	382	**526**	**648**	**780**	**896**
28	**94**	**94**	174	**188**	258	**302**	339	**427**	406	**561**	**693**	**826**	**953**
29	94	**98**	174	**196**	269	**322**	346	**452**	417	**593**	**733**	**876**	**1015**
30	**105**	102	194	**209**	310	**335**	390	**473**	458	**620**	**787**	**925**	**1086**
31	**107**	106	212	**215**	310	**347**	390	**504**	482	**657**	**812**	**976**	**1138**
32	**114**	110	212	**228**	341	**371**	419	**529**	518	**695**	**866**	**1026**	**1194**
33	**114**	113	215	**234**	341	**379**	430	**548**	537	**723**	**900**	**1084**	**1253**
34	**133**	117	230	**244**	358	**403**	470	**586**	561	**751**	**945**	**1135**	**1329**
35	**133**	122	234	**255**	366	**418**	495	**601**	586	**811**	**1002**	**1190**	**1398**
36	**133**	129	250	**261**	400	**427**	518	**631**	636	**831**	**1042**	**1257**	**1459**
37	**152**	131	266	**275**	408	**454**	528	**648**	668	**863**	**1079**	**1300**	**1516**
38	**152**	134	**283**	279	415	**464**	561	**681**	709	**923**	**1127**	**1367**	**1597**
39	**152**	139	283	**290**	439	**486**	561	**706**	726	**938**	**1192**	**1434**	**1665**
40	**155**	146	291	**301**	492	**505**	632	**739**	786	**970**	**1224**	**1489**	**1742**
41	**162**	147	293	**309**	492	**525**	632	**776**	802	**1016**	**1271**	**1562**	**1820**
42	**168**	152	319	**325**	496	**543**	670	**791**	903	**1064**	**1333**	**1639**	**1920**
43	**168**	157	323	**329**	520	**558**	670	**830**	903	**1112**	**1377**	**1683**	**1973**
44	**186**	161	331	**349**	548	**582**	696	**862**	903	**1140**	**1463**	**1752**	**2072**
45	**186**	166	347	**362**	565	**615**	737	**891**	926	**1192**	**1480**	**1820**	**2130**
46	**189**	169	366	**370**	592	**615**	797	**918**	985	**1243**	**1548**	**1906**	**2208**
47	**189**	173	**378**	**378**	611	**634**	797	**940**	985	**1268**	**1616**	**1958**	**2331**
48	**189**	178	**413**	385	632	**673**	857	**976**	1054	**1325**	**1658**	**2017**	**2387**
49	**196**	180	**415**	399	634	**680**	893	**1015**	1074	**1356**	**1729**	**2103**	**2470**
50	**213**	185	**415**	414	663	**699**	893	**1042**	1113	**1397**	**1772**	**2179**	**2556**
51	**213**	189	421	**426**	692	**727**	917	**1067**	1161	**1450**	**1855**	**2243**	**2639**
52	**213**	198	**455**	429	709	**742**	1003	**1100**	1231	**1486**	**1888**	**2325**	**2745**
53	**216**	200	**455**	447	716	**765**	1003	**1136**	1241	**1537**	**1949**	**2429**	**2825**
54	**233**	213	**477**	454	760	**783**	1019	**1171**	1288	**1577**	**2006**	**2473**	**2892**
55	**243**	214	**483**	477	760	**805**	1082	**1198**	1325	**1639**	**2084**	**2570**	**3054**
56	**243**	216	**515**	479	784	**830**	1104	**1236**	1358	**1701**	**2162**	**2623**	**3100**
57	**261**	221	**515**	490	846	**854**	1136	**1265**	1479	**1721**	**2194**	**2704**	**3215**
58	**261**	227	**539**	500	846	**878**	1166	**1303**	1479	**1795**	**2258**	**2796**	**3305**
59	**266**	229	**544**	519	849	**905**	1223	**1328**	1509	**1821**	**2356**	**2881**	**3399**
60	**273**	237	**568**	530	904	**928**	1242	**1381**	1577	**1899**	**2393**	**2939**	**3500**
61	**274**	244	**620**	538	904	**939**	1258	**1413**	1615	**1928**	**2488**	**3021**	**3588**
62	**283**	245	**620**	554	934	**991**	1306	**1450**	1680	**2023**	**2541**	**3132**	**3700**
63	**297**	249	**620**	575	967	**989**	1380	**1497**	1680	**2035**	**2607**	**3215**	**3767**
64	**297**	258	**625**	579	985	**1009**	1430	**1526**	1769	**2093**	**2734**	**3292**	**3955**
65	**314**	260	**630**	582	997	**1035**	1430	**1565**	1786	**2132**	**2723**	**3357**	**4034**
66	**314**	269	**666**	602	1050	**1051**	1476	**1590**	1857	**2180**	**2841**	**3474**	**4143**
67	**314**	270	**666**	614	1072	**1085**	1482	**1646**	1868	**2238**	**2868**	**3543**	**4224**
68	**314**	278	**685**	623	1087	**1119**	1538	**1664**	1940	**2295**	**2956**	**3647**	**4360**
69	**324**	280	**698**	650	1112	**1114**	1588	**1704**	1965	**2351**	**3075**	**3716**	**4455**
70	**325**	285	**716**	658	**1150**	1135	1633	**1759**	2130	**2417**	**3130**	**3841**	**4539**
71	**325**	289	**716**	665	1150	**1187**	1644	**1783**	2130	**2451**	**3161**	**3936**	**4689**
72	**341**	296	**750**	678	**1203**	1197	1768	**1862**	2177	**2503**	**3220**	**4027**	**4812**
73	**350**	299	**759**	688	1229	**1242**	1768	**1872**	2206	**2598**	**3305**	**4134**	**4873**
74	**350**	306	**767**	703	1229	**1269**	1774	**1910**	2244	**2614**	**3432**	**4224**	**5038**
75	**350**	310	**771**	714	1274	**1282**	1862	**1963**	2295	**2703**	**3513**	**4298**	**5171**
76	**363**	324	**813**	750	1300	**1318**	1935	**2024**	2375	**2756**	**3559**	**4395**	**5254**
77	**363**	325	**823**	762	1308	**1331**	1947	**2051**	2403	**2819**	**3617**	**4492**	**5399**
78	**387**	337	**844**	761	**1382**	1360	2014	**2079**	2505	**2870**	**3684**	**4577**	**5489**

Table 3: (Maximin) squared ℓ^2-distance found using periodic designs (PD) and simulated annealing (SA).

n	3 dim		4 dim		5 dim		6 dim		7 dim		8 dim	9 dim	10 dim
	PD	SA	PD	SA	PD	SA	PD	SA	PD	SA	SA	SA	SA
79	**387**	333	**848**	788	1382	**1399**	2037	**2120**	2525	**2950**	**3775**	**4705**	**5633**
80	**403**	344	**873**	786	1395	**1430**	2037	**2152**	2590	**2979**	**3877**	**4807**	**5773**
81	**406**	338	**916**	782	1406	**1431**	2064	**2217**	2642	**3086**	**4001**	**4888**	**5901**
82	**406**	353	**938**	825	1475	**1482**	2141	**2239**	2753	**3118**	**3998**	**5030**	**6013**
83	**417**	369	**940**	829	1501	**1509**	2141	**2290**	2767	**3195**	**4076**	**5102**	**6097**
84	**426**	363	**967**	838	**1534**	1510	2229	**2325**	2838	**3227**	**4183**	**5222**	**6273**
85	**426**	369	**967**	877	1552	**1566**	2232	**2399**	2874	**3299**	**4324**	**5340**	**6397**
86	**428**	376	**967**	867	1573	**1578**	2375	**2437**	3103	**3335**	**4397**	**5423**	**6491**
87	**428**	374	**976**	877	**1598**	1589	2375	**2476**	3103	**3450**	**4474**	**5538**	**6622**
88	**437**	374	**1050**	890	**1685**	1629	2398	**2513**	3183	**3500**	**4524**	**5667**	**6803**
89	**443**	378	**1050**	907	**1690**	1654	2400	**2562**	3183	**3541**	**4578**	**5774**	**6872**
90	**481**	384	**1060**	940	**1710**	1696	2516	**2633**	3190	**3661**	**4699**	**5832**	**7040**
91	**481**	393	**1089**	951	**1748**	1724	2516	**2674**	3234	**3677**	**4850**	**5969**	**7163**
92	**481**	394	**1089**	966	**1805**	1750	2599	**2729**	3277	**3760**	**4873**	**6081**	**7286**
93	**481**	402	**1098**	962	**1813**	1795	2604	**2726**	3361	**3811**	**4984**	**6231**	**7488**
94	**481**	405	**1124**	986	**1881**	1811	2747	**2788**	3474	**3888**	**5067**	**6329**	**7536**
95	**481**	413	**1135**	1010	**1901**	1846	2747	**2817**	3531	**3940**	**5154**	**6396**	**7741**
96	**509**	414	**1261**	1023	**1965**	1863	2769	**2911**	3639	**4070**	**5220**	**6516**	**7777**
97	**515**	419	**1261**	1027	**1965**	1899	2817	**2960**	3639	**4069**	**5316**	**6649**	**8038**
98	**531**	429	**1261**	1055	**1965**	1929	2850	**3001**	3690	**4147**	**5445**	**6776**	**8242**
99	**531**	449	**1261**	1040	**2009**	1950	2878	**3043**	3731	**4214**	**5477**	**6912**	**8344**
100	**554**	451	**1261**	1074	**2053**	1975	3000	**3117**	3903	**4335**	**5597**	**6983**	**8450**

Table 3: (Maximin) squared ℓ^2-distance found using periodic designs (PD) and simulated annealing (SA) (*continued*).

References

Aarts, E.H.L. and J.K. Lenstra (1997). *Local search in combinatorial optimization*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Chichester: John Wiley & Sons.

Alam, F.M., K.R. McNaught, and T.J. Ringrose (2004). A comparison of experimental designs in the development of a neural network simulation metamodel. *Simulation Modelling: Practice and Theory*, **12 (7–8)**, 559–578.

Baer, D. (1992). Punktverteilungen in Würfeln beliebiger Dimension bezüglich der Maximum-norm. *Wiss. Z. Pädagog. Hochsch. Erfurt/Mühlhausen, Math.-Naturwiss. Reihe*, **28**, 87–92.

Barthelemy, J.F.M. and R.T. Haftka (1993). Approximation concepts for optimum structural design – A review. *Structural Optimization*, **5(3)**, 129–144.

Bates, R.A., R.J. Buck, E. Riccomagno, and H.P. Wynn (1996). Experimental design and observation for large systems. *Journal of the Royal Statistical Society: Series B*, **58**, 77–94.

Bates, S.J., J. Sienz, and V.V. Toropov (2004). Formulation of the optimal Latin hypercube design of experiments using a permutation genetic algorithm. *AIAA 2004-2011*, 1–7.

Bursztyn, D. and D.M. Steinberg (2006). Comparison of designs for computer experiments. *Journal of Statistical Planning and Inference*, **136(3)**, 1103–1119.

Crary, S.B. (2001). WebDOE$^{\text{TM}}$. `http://www.webdoe.cc`.

Crary, S.B. (2002). Design of computer experiments for metamodel generation. *Analog Integrated Circuits and Signal Processing*, **32(1)**.

Crary, S.B., P. Cousseau, D. Armstrong, D. M. Woodcock, E.H. Mok, O. Dubochet, P. Lerch, and P. Renaud (2000). Optimal design of computer experiments for metamodel generation using I-OPT$^{\text{TM}}$. *Computer Modeling in Engineering & Sciences*, **1(1)**, 127–139.

Dam, E.R. van (2005). Two-dimensional minimax Latin hypercube designs. *CentER Discussion Paper 2005-105*. Tilburg University.

Dam, E.R. van, B.G.M. Husslage, D. den Hertog, and J.B.M. Melissen (2006). Maximin Latin hypercube designs in two dimensions. *Operations Research*. To appear.

Dimnaku, A., R. Kincaid, and M.W. Trosset (2005). Approximate solutions of continuous dispersion problems. *Annals of Operations Research*, **136(1)**, 65–80.

Driessen, L.T., H.P. Stehouwer, and J.J. Wijker (2002). Structural mass optimization of the engine frame of the Ariane 5 ESC-B. *Proceedings of the European Conference on Spacecraft, Structures, Materials & Mechanical Testing*. Toulouse, France.

Erkut, E. (1990). The discrete para-dispersion problem. *European Journal of Operational Research*, **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. *Sitzungsber., Abt. II, Österr. Akad. Wiss., Math.-Naturwiss. Kl.*, **198**, 27–44.

Giunta, A.A., S.F. Wojtkiewicz, and M.S. Eldred (2003). Overview of modern design of experiments methods for computational simulations. *AIAA 2003-0649*, 1–17.

Hertog, D. den and H.P. Stehouwer (2002). Optimizing color picture tubes by high-cost nonlinear programming. *European Journal of Operational Research*, **140(2)**, 197–211.

Jin, R., W. Chen, and A. Sudjianto (2005). An efficient algorithm for constructing optimal design of computer experiments. *Journal of Statistical Planning and Inference*, **134(1)**, 268–287.

Johnson, M.E., L.M. Moore, and D. Ylvisaker (1990). Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, **26**, 131–148.

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

Kirkpatrick, S., C.D. Gelatt, and M.P. Vecchi (1983). Optimization by simulated annealing. *Science*, **220**, 671–680.

Koehler, J.R. and A.B. Owen (1996). *Computer experiments*, Volume 13 of *Handbook of Statistics*. Elsevier Science B.V.

Locatelli, M. and U. Raber (2002). Packing equal circles in a square: A deterministic global optimization approach. *Discrete Applied Mathematics*, **122(1-3)**, 139–166.

Markót, M.Cs. and T. Csendes (2005). A new verified optimization technique for the "packing circles in a unit square" problems. *SIAM Journal on Optimization*, **16(1)**, 193–219.

McKay, M.D., R. J. Beckman, and W.J. Conover (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, **16**, 239–245.

Melissen, J.B.M. (1997). *Packing and covering with circles*. Ph. D. thesis, Utrecht University.

Mladenovic, N., F. Plastria, and D. Urosevic (2005). Reformulation descent applied to circle packing problems. *Computers & Operations Research*, **32(9)**, 2419–2434.

Morris, M.D. and T.J. Mitchell (1995). Exploratory designs for computer experiments. *Journal of Statistical Planning and Inference*, **43**, 381–402.

Nurmela, K.J. and P.R.J. Östergård (1999). More optimal packings of equal circles in a square. *Discrete and Computational Geometry*, **22**, 439–547.

Peikert, R., D. Würtz, M. Monagan, and C. den Groot (1991). Packing circles in a sphere: A review and new results. *Proceedings of the 15th IFIP Conference on System Modeling and Optimization, Springer Lecture Notes in Control and Information Sciences*, **180**, 111–124.

Rikards, R. and J. Auzins (2004). Response surface method for solution of structural identification problems. *Inverse Problems in Engineering*, **12(1)**, 59–70.

Santner, Th.J., B.J. Williams, and W.I. Notz (2003). *The design and analysis of computer experiments*. Springer Series in Statistics. New York: Springer-Verlag.

Simpson, T.W., A.J. Booker, D. Ghosh, A.A. Giunta, P.N. Koch, and R.-J. Yang (2004). Approximation methods in multidisciplinary analysis and optimization: A panel discussion. *Structural and Multidisciplinary Optimization*, **27(5)**, 302–313.

Simpson, T.W., D.K.J. Lin, and W. Chen (2001). Sampling strategies for computer experiments: Design and analysis. *International Journal of Reliability and Applications*, **2(3)**, 209–240.

Sobieszczanski-Sobieski, J. and R.T. Haftka (1997). Multidisciplinary aerospace design optimization: Survey of recent developments. *Structural Optimization*, **14(1)**, 1–23.

Specht, E. (2005). Packomania. `http://www.packomania.com`, July 2005.

Stinstra, E.D., D. den Hertog, H.P. Stehouwer, and A. Vestjens (2003). Constrained maximin designs for computer experiments. *Technometrics*, **45(4)**, 340–346.

Trosset, M.W. (1999). Approximate maximin distance designs. *Proceedings of the Section on Physical and Engineering Sciences*, 223–227.

Ye, K.Q., W. Li, and A. Sudjianto (2000). Algorithmic construction of optimal symmetric Latin hypercube designs. *Journal of Statistical Planning and Inference*, **90(1)**, 145–159.