CentER

Discussion Paper

No. 2004–66

ONE-DIMENSIONAL NESTED MAXIMIN DESIGNS

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

July 2004

ISSN 0924-7815

TILBURG UNIVERSITY

One-dimensional nested maximin designs

Edwin van Dam[1] • Bart Husslage[2] • Dick den Hertog

Department of Econometrics and Operations Research, Tilburg University
P.O. Box 90153, 5000 LE Tilburg, The Netherlands
center.uvt.nl/staff/dam/ • *center.uvt.nl/phd_stud/husslage/* • *center.uvt.nl/staff/hertog/*

Abstract

The design of computer experiments is an important step in black box evaluation and optimization processes. When dealing with multiple black box functions the need often arises to construct designs for all black boxes jointly, instead of individually. These so-called nested designs are used to deal with linking parameters and sequential evaluations. In this paper we discuss one-dimensional nested maximin designs. We show how to nest two designs optimally and develop a heuristic to nest three and four designs. Furthermore, it is proven that the loss in space-fillingness, with respect to traditional maximin designs, is at most 14.64 percent and 19.21 percent, when nesting two and three designs, respectively.

Keywords: Maximin design, space-filling, linking parameter, black box, computer simulation, mixed integer programming.
JEL Classification: C90.

1 Introduction

Maximin designs play an important role in the field of (deterministic) black box evaluation and optimization. By nature, a black box function is not given explicitly, however, we may perform function evaluations. Based on these evaluations an approximation model for the black box can be constructed, gaining insight in the black box and opening the way for optimization techniques. Unfortunately, function evaluations often constitute time-consuming computer simulations, thereby limiting the number of evaluations performed. A proper design of computer experiments then becomes vitally important. See, e.g. [1], [2], [4], [7], and [8].

We will use the term *design* to denote the set of points that will be evaluated. Such a design should at least be *space-filling* in some sense to provide information about the entire black box domain. Several space-filling measures are used in the literature, like minimax, maximin, IMSE, and maximum entropy. See, e.g. [5], [6], and [8]. A good survey of these criteria can be found in [9], in which it is also shown that the maximin measure, which maximizes the minimal distance among all pairs of points, is preferable to the other criteria when conducting computer experiments. Therefore, we will use the maximin criterion in this paper.

In real-life problems there is often a need for *nested maximin designs*. Consider the case where we want to construct two maximin designs, such that one design is a subset of the other design. Due to this restriction the resulting designs will be nested and we will therefore call the combination of the two designs a nested maximin design. Such nested maximin designs can be found by mixed integer programming, however, this takes a lot of computation time. This paper shows how to construct one-dimensional nested maximin designs. Furthermore, it is proven that the loss in space-fillingness of this type of maximin design is at most 14.64% and 19.21%, when nesting two and three designs, respectively. There are two main reasons for nesting maximin designs: *linking parameters* and *sequential evaluations*.

To start with the first; consider a product that consists of two components, each of them represented by a black box function. In practice it often occurs that the functions have an input parameter in common, also called a *linking parameter*, see [3]. Evaluating such a linking parameter at the same setting in both functions (i.e. component-wise) leads to an evaluation of the product. Not only do product evaluations provide a better understanding of the product, they are also very useful in the product optimization

[1] The research of E.R. van Dam has been made possible by a fellowship of the Royal Netherlands Academy of Arts and Sciences.
[2] The research of B.G.M. Husslage is funded by the SamenwerkingsOrgaan Brabantse Universiteiten (SOBU).

process. Another reason for using the same settings for (linking) parameters is due to physical restrictions on the simulation tools. Setting the parameters for computer experiments can be a time-consuming job in practice, since characteristics, like shape and structure, have to be redefined for every new experiment. Therefore, it is preferable to use the same settings as much as possible. By constructing nested maximin designs we can determine the settings for linking parameters.

Sequential evaluations are a second reason for using nested maximin designs. In practice it is common that after evaluating an initial set of points, extra evaluations are needed. As an example, suppose we construct an approximation model for a black box function based on n_1 function evaluations. However, after validating the obtained model it turns out that an extra set of, say, $n_2 - n_1$ function evaluations is needed to build a proper model. We then face the problem of constructing a maximin design on n_2 points, given the initial maximin design on n_1 points. It would be better to anticipate on the possibility of extra evaluations. This can be accomplished by constructing the two maximin designs (on n_1 and n_2 points) at once, hence, by constructing a nested maximin design.

We will now give a more strict formulation of our problem. Let there be $m \in \mathbb{N}$ nested sets $X_1 \subseteq X_2 \subseteq \cdots \subseteq X_m$ and index sets $I_1 \subseteq I_2 \subseteq \cdots \subseteq I_m = \{1, \ldots, n_m\}$, where $X_i = \{x_j | j \in I_i\}$ and $|I_i| = n_i$, $i = 1, \ldots, m$. Thus I_i tells us which x_j are contained in set X_i, and the X_i define the nested design. We assume without loss of generality that all points $x_j \in [0, 1]$. Note that when we consider a set X_i independently, a space-filling distribution of the x_j over the $[0, 1]$ interval is obtained by spreading the points equidistantly over the interval, resulting in a minimal distance of $\frac{1}{n_i - 1}$ among the points. Our aim is to determine x_j and I_i such that every set X_i is as much as possible space-filling with respect to the maximin criterion. To this end we define d_i as the minimal scaled distance among all points in the set X_i, i.e. $d_i = \min_{j,k \in I_i,\, j \neq k} (n_i - 1)|x_j - x_k|$ for all i. Then we have to maximize $d = \min_i d_i$ over all $I_1 \subseteq I_2 \subseteq \cdots \subseteq I_m$, with $|I_i| = n_i$, and $x_j \in [0, 1]$. This will yield the maximin distance d and a corresponding nested maximin design in terms of the I_i and x_j.

This paper is organized as follows. In Section 2 we derive an exact formula for the maximin distance of two nested sets. This derivation also shows how to construct the corresponding nested maximin designs. In Section 3 we continue with three nested sets, for which we prove a lower bound on the maximin distance and develop a heuristic to construct good nested designs. Section 4 shows that the heuristic for three nested sets can also be used to construct good nested designs for four nested sets. Furthermore, in this section we prove a lower bound on the maximin distance for all $m \in \mathbb{N}$ that fulfill the restriction $n_m < 2n_1$. Finally, Section 5 gives the conclusions and some topics for further research.

2 Two nested sets

We first discuss the case of two nested sets, i.e. $m = 2$. In Section 2.1 we start with the general problem formulation and show how to nest two sets optimally. Furthermore, in this section we derive a formula for the maximin distance and prove a tight lower bound on this distance. In Section 2.2 we introduce the notion of dominance and discuss the trade-off between d_1 and d_2.

2.1 Maximin distance

The general problem for two nested sets can be formalized as the following mathematical program:

$$\begin{array}{lll} \max & \min\limits_{\substack{j,k \in I_i \\ i=1,2;\, j \neq k}} & (n_i - 1)|x_j - x_k| \\ \text{s.t.} & I_1 \subseteq I_2 & \\ & |I_1| = n_1 & \\ & 0 \leq x_j \leq 1, \quad j \in I_2. & \end{array} \tag{1}$$

To obtain a feasible solution that maximizes the objective function in (1), we may choose without loss of generality $x_1 = 0$, $x_{n_2} = 1$, $x_i < x_{i+1}$, $1 \in I_1$, and $n_2 \in I_1$. For a given I_1, containing the indices, say, $1 = a_1 < a_2 < \cdots < a_{n_1} = n_2$ we introduce the sequence $v = (v_1, \ldots, v_{n_1 - 1})$ given by $v_i = a_{i+1} - a_i$. Thus $v_i - 1$ gives the number of additional points of X_2 between the i-th and $(i+1)$-st point of X_1. It is clear that the set of possible I_1 is in one-one correspondence to the set of positive integral sequences v, summing to $n_2 - 1$. Now the approach to solve problem (1) is to first fix I_1, and its corresponding $a = (a_1, \ldots, a_{n_1})$ and v, and obtain an expression for the maximal distance δ_v, subject to the remaining constraints, and then to maximize δ_v over all v. It turns out that finding δ_v is rather simple.

Lemma 1 *For fixed I_1, and corresponding a, v, the optimal value δ_v equals*

$$(\sum_{i=1}^{n_1-1} \max\{\frac{v_i}{n_2-1}, \frac{1}{n_1-1}\})^{-1}.$$

Proof. Fix a, v, and let δ_v be the corresponding maximal distance. Since $x_{i+1} - x_i \geq \frac{\delta_v}{n_2-1}$ for all i, we have that $x_{a_{i+1}} - x_{a_i} \geq v_i \frac{\delta_v}{n_2-1}$. We also have that $x_{a_{i+1}} - x_{a_i} \geq \frac{\delta_v}{n_1-1}$, hence

$$x_{a_{i+1}} - x_{a_i} \geq \max\{v_i \frac{\delta_v}{n_2-1}, \frac{\delta_v}{n_1-1}\}.$$

From this we find that $1 = x_{a_{n_1}} - x_{a_1} \geq \delta_v \sum_{i=1}^{n_1-1} \max\{\frac{v_i}{n_2-1}, \frac{1}{n_1-1}\}$, which shows that the stated expression for δ_v is an upper bound. It is clear from the above that, and how, this upper bound can be attained, which proves the lemma. $\square$

We now have to maximize δ_v over all appropriate sequences v. For ease of notation define $c_2 = \frac{n_2-1}{n_1-1}$.

Proposition 1 *Let $2 \leq n_1 \leq n_2$. The maximin distance in (1) is given by*

$$d = \frac{1}{1 + \lfloor c_2 \rfloor + \lceil c_2 \rceil - c_2 - \lfloor c_2 \rfloor \lceil c_2 \rceil \frac{1}{c_2}}. \tag{2}$$

Proof. As mentioned before, we have to maximize δ_v, which is equivalent to minimizing

$$\sum_{i=1}^{n_1-1} \max\{\frac{v_i}{n_2-1}, \frac{1}{n_1-1}\}$$

over all integer-valued v, such that $\sum_{i=1}^{n_1-1} v_i = n_2 - 1$.

We claim that it is optimal to let v only take values $\lfloor c_2 \rfloor$ and $\lceil c_2 \rceil$. This is clearly true if $n_2 - 1$ is a multiple of $n_1 - 1$, since in that case picking a larger value than c_2 for any of the v_i will increase the objective function. Therefore, assume now that $n_2 - 1$ is not a multiple of $n_1 - 1$. To prove our claim, first assume that $v_i \leq \lfloor c_2 \rfloor - 1$ for some i. Let j be such that $v_j \geq \lceil c_2 \rceil$ (such a j exists). Then by adding 1 to v_i, and subtracting 1 from v_j, we obtain v' for which the objective function is strictly smaller than for v. This follows from the inequality $\max\{\frac{v_i}{n_2-1}, \frac{1}{n_1-1}\} + \max\{\frac{v_j}{n_2-1}, \frac{1}{n_1-1}\} > \max\{\frac{v_i+1}{n_2-1}, \frac{1}{n_1-1}\} + \max\{\frac{v_j-1}{n_2-1}, \frac{1}{n_1-1}\}$, which is easily checked to be true. Hence, the original v is not optimal. Similarly, the case where $v_i \geq \lceil c_2 \rceil + 1$ for some i is ruled out.

Thus it follows that the optimal v has $v_i = \lfloor c_2 \rfloor$ for $p = (n_1 - 1)(\lceil c_2 \rceil - c_2)$ values of i, and $v_i = \lceil c_2 \rceil$ for the remaining i. The value for d now easily follows from Lemma 1. $\square$

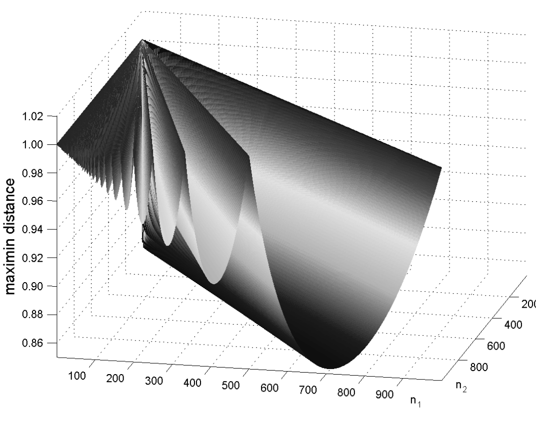

Figure 1: Maximin distance as function of n_1 and n_2.

For a graphical representation of the maximin distance as function of n_1 and n_2, see Figure 1. Using above results, a nested maximin design, with maximin distance as in (2), can easily be constructed:

$$x_{j+1} = \begin{cases} \frac{d}{n_1-1}\frac{j}{\lfloor c_2 \rfloor} & j = 0, \ldots, p\lfloor c_2 \rfloor; \\ \frac{d}{n_1-1}p + \frac{d}{n_2-1}(j - p\lfloor c_2 \rfloor) & j = p\lfloor c_2 \rfloor + 1, \ldots, n_2 - 1; \end{cases}$$
$$(I_1)_{j+1} = \begin{cases} 1 + j\lfloor c_2 \rfloor & j = 0, \ldots, p; \\ 1 + p\lfloor c_2 \rfloor + (j-p)\lceil c_2 \rceil & j = p+1, \ldots, n_1 - 1. \end{cases} \tag{3}$$

As an example, we construct a nested maximin design for $n_1 = 4$ and $n_2 = 8$. From (2) we get that the maximin distance equals $d = \frac{21}{23} \simeq 0.9130$. Substituting d and $p = 2$ in (3) results in the points $x_1 = 0$, $x_2 = \frac{7}{46}$, $x_3 = \frac{14}{46}$, $x_4 = \frac{21}{46}$, $x_5 = \frac{28}{46}$, $x_6 = \frac{34}{46}$, $x_7 = \frac{40}{46}$, and $x_8 = 1$, and yields the set $I_1 = \{1, 3, 5, 8\}$, implying that $X_1 = \{x_1, x_3, x_5, x_8\}$. See Figure 2 for a graphical representation of this nested maximin design.

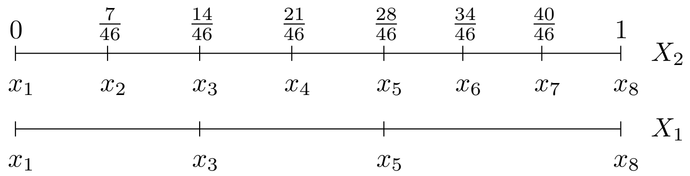

Figure 2: A nested maximin design for $n_1 = 4$ and $n_2 = 8$, with $d = \frac{21}{23} \simeq 0.9130$.

Besides computing the maximin distance for a given n_1 and n_2, (2) can also be used to prove a general lower bound on the maximin distance.

Proposition 2 *Let* $2 \leq n_1 \leq n_2$*. Then* $1 \geq d > (4 - 2\sqrt{2})^{-1} \simeq 0.853553$.

Proof. Consider the function $z : [1, \infty) \to \mathbb{R}$ given by

$$z(c_2) = 1 + \lfloor c_2 \rfloor + \lceil c_2 \rceil - c_2 - \lfloor c_2 \rfloor \lceil c_2 \rceil \frac{1}{c_2} = 1 + \frac{(c_2 - \lfloor c_2 \rfloor)(\lceil c_2 \rceil - c_2)}{c_2} \geq 1.$$

If $c_2 \in \mathbb{N}$ then $z(c_2) = 1$, i.e. z is minimal and hence $d = z(c_2)^{-1} \leq 1$, else

$$z(c_2 + 1) = 1 + \frac{(c_2 - \lfloor c_2 \rfloor)(\lceil c_2 \rceil - c_2)}{c_2 + 1} < z(c_2); \quad c_2 \notin \mathbb{N}.$$

Therefore, in this case z is maximal for some $c_2 \in (1, 2)$. Restrict z to $(1, 2)$:

$$z(c_2) = 1 + 1 + 2 - c_2 - \frac{2}{c_2} = 4 - c_2 - \frac{2}{c_2},$$

which is maximal for $c_2 = \sqrt{2}$. For $c_2 \in \mathbb{Q}$, $c_2 \geq 1$:

$$z(c_2) < z(\sqrt{2}) = 4 - 2\sqrt{2}, \text{ and then } d > \frac{1}{z(\sqrt{2})} = \frac{1}{4 - 2\sqrt{2}} = \frac{1}{2} + \frac{1}{4}\sqrt{2} \simeq 0.853553. \quad \square$$

Note that the obtained lower bound is tight since we can take c_2 arbitrarily close to $\sqrt{2}$. The interpretation of this lower bound is that for all values of n_1 and n_2, by nesting the sets X_1 and X_2 we will never lose more than 14.64%, with respect to the "restriction free" maximin distance. In practice this implies that a linking parameter can be included in the maximin designs, at a cost of using designs that are at most 14.65% worse with respect to space-fillingness.

In case of sequential evaluations the interpretation is somewhat different. A standard way to perform (two-stage) sequential evaluations is to first choose n_1 points, equidistantly distributed over the interval $[0, 1]$. After the evaluations, if needed, $n_2 - n_1$ extra points are taken, resulting in $d' = \frac{c_2}{\lceil c_2 \rceil}$; see Section 2.2. Clearly, $d \geq d'$ and $d' = \frac{c_2}{\lceil c_2 \rceil} \geq \frac{c_2}{c_2 + 1} > \frac{1}{2}$, for $c_2 > 1$. If one evaluation stage turns out to be sufficient, using the points in (3) will result in a design that is at most 14.65% worse than the (standard) equidistant design (since we lose $1 - d$). However, if a second evaluation stage is needed then our approach results in a better space-filling design (since we win $d - d'$). Figure 3 shows the net gain of our approach, i.e. $(d - 1) + (d - d')$, as function of c_2. For $n_2 \leq 100$ the net gain takes values in the interval $[-0.07, 0.48]$.

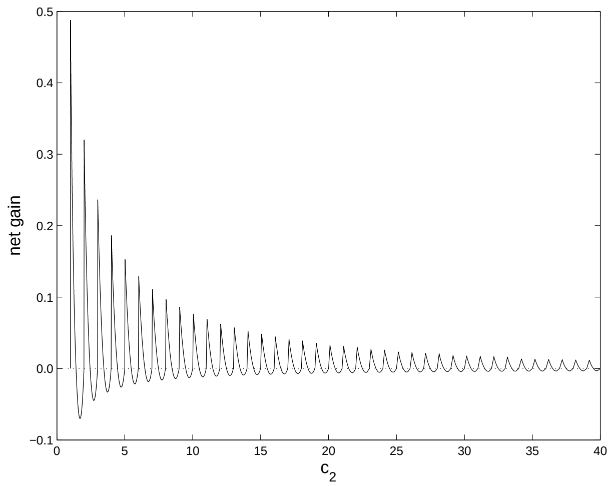

Figure 3: Net gain of our approach as function of c_2.

2.2 Dominance

In the last section we appraised the sets X_1 and X_2 to be equally important. What if one set is more important than the other? Or, given a fixed value for d_1, what is the corresponding maximal value of d_2? To examine this, we first introduce the notion of *dominance*. We will call a combination (d_1, d_2) dominant if it is not possible to improve one of the coordinates, without deteriorating the other coordinate. Knowing the dominant combinations is very useful in practice. It enables us to determine the trade-off between d_1 and d_2, i.e. it helps us finding a combination that best satisfies our requirements, like "X_2 is twice as important as X_1". Note that the maximin combination (d, d), with d as in (2), is dominant. The combinations $(1, \frac{c_2}{\lceil c_2 \rceil})$ and $(\frac{\lfloor c_2 \rfloor}{c_2}, 1)$ are also dominant, which can be argued as follows:

- Fixing $d_1 = 1$, the points of X_1 must be equidistantly distributed, i.e. $X_1 = \{0, \frac{1}{n_1-1}, \frac{2}{n_1-1}, \ldots, 1\}$. Due to the restriction $I_1 \subseteq I_2$ we need to find settings for the $n_2 - n_1$ extra points in X_2, such that d_2 is maximal. This is accomplished by choosing these $n_2 - n_1$ points as equally as possible spread over the $n_1 - 1$ intervals formed by the points in X_1, which corresponds to v taking only the values $\lfloor c_2 \rfloor$ and $\lceil c_2 \rceil$, as before. Hence, after scaling, this gives a distance of $d_2 = (n_2 - 1)\frac{1}{n_1-1}\lceil c_2 \rceil^{-1} = \frac{c_2}{\lceil c_2 \rceil}$.
- Fixing $d_2 = 1$, the points of X_2 must be equidistantly distributed, i.e. $X_2 = \{0, \frac{1}{n_2-1}, \frac{2}{n_2-1}, \ldots, 1\}$. To maximize d_1, the $n_2 - 1$ intervals must as equally as possible be spread over the $n_1 - 1$ intervals that are to be formed by the points in X_1. Every interval of X_1 will then contain either $\lfloor c_2 \rfloor$ or $\lceil c_2 \rceil$ intervals of length $\frac{1}{n_2-1}$, and the distance, after scaling, will be given by $d_1 = (n_1 - 1)\frac{1}{n_2-1}\lfloor c_2 \rfloor = \frac{\lfloor c_2 \rfloor}{c_2}$.

Since $(1, \frac{c_2}{\lceil c_2 \rceil})$ and $(\frac{\lfloor c_2 \rfloor}{c_2}, 1)$ bound the values of d_1 and d_2 we will call them extreme dominant combinations. Moreover, note that these bounds imply that $d \geq \max\{\frac{c_2}{\lceil c_2 \rceil}, \frac{\lfloor c_2 \rfloor}{c_2}\}$. For a given n_1 and n_2 all dominant combinations can be characterized by the following linear function.

Proposition 3 *Let* $2 \leq n_1 \leq n_2$. *All dominant combinations* (d_1, d_2) *are characterized by the linear function* $f : [\frac{\lfloor c_2 \rfloor}{c_2}, 1] \rightarrow [\frac{c_2}{\lceil c_2 \rceil}, 1]$, *where*

$$d_2 = f(d_1) = ((c_2 - \lceil c_2 \rceil)d_1 + 1)\frac{c_2}{\lceil c_2 \rceil(c_2 - \lfloor c_2 \rfloor)}. \tag{4}$$

Proof. Like in Lemma 1, we have for a given I_1, and corresponding a, v, that

$$1 \geq \sum_{i=1}^{n_1-1} \max\{\frac{v_i}{n_2-1}d_2, \frac{1}{n_1-1}d_1\}. \tag{5}$$

Hence, for a given a, v, and $d_1 \leq 1$, it is optimal to choose d_2 as large as possible, such that equality is attained in (5).

We claim that for any d_1, with $\frac{\lfloor c_2 \rfloor}{c_2} \le d_1 \le 1$, a maximal d_2 is obtained by letting v take only the values $\lfloor c_2 \rfloor$ and $\lceil c_2 \rceil$, just like in Proposition 1. Note that this needs no further proof for $d_1 = \frac{\lfloor c_2 \rfloor}{c_2}$ and $d_1 = 1$, therefore we may assume that $\frac{\lfloor c_2 \rfloor}{c_2} < d_1 < 1$, and, hence, that c_2 is not an integer.

To prove the claim, fix d_1, and suppose that there is a v giving an optimal d_2 with $v_i \ge \lceil c_2 \rceil + 1$ for some i. Let j be such that $v_j \le \lfloor c_2 \rfloor$ (such a j exists). Since d_2 is optimal we may assume that $d_2 \ge \frac{c_2}{\lceil c_2 \rceil}$. Now let v' be obtained from v by subtracting 1 from v_i, and adding 1 to v_j. Since d_2 is optimal, the d_2' corresponding to v' is at most d_2.

From the equalities in (5) for the pairs (v, d_2) and (v', d_2'), and the inequality $d_2' \le d_2$, we obtain that $\max\{\frac{v_i}{n_2-1}d_2, \frac{1}{n_1-1}d_1\} + \max\{\frac{v_j}{n_2-1}d_2, \frac{1}{n_1-1}d_1\} \le \max\{\frac{v_i-1}{n_2-1}d_2, \frac{1}{n_1-1}d_1\} + \max\{\frac{v_j+1}{n_2-1}d_2, \frac{1}{n_1-1}d_1\}$. Because of the inequalities $v_i \ge \lceil c_2 \rceil + 1$, $v_j \le \lfloor c_2 \rfloor$, $1 \ge d_2 \ge \frac{c_2}{\lceil c_2 \rceil}$, and $\frac{\lfloor c_2 \rfloor}{c_2} < d_1 < 1$, this reduces to $\frac{v_i}{n_2-1}d_2 + \frac{1}{n_1-1}d_1 \le \frac{v_i-1}{n_2-1}d_2 + \max\{\frac{v_j+1}{n_2-1}d_2, \frac{1}{n_1-1}d_1\}$. Now this implies that $\frac{v_j+1}{n_2-1}d_2 \ge \frac{1}{n_1-1}d_1$, and, hence, the inequality further reduces to $\frac{1}{n_1-1}d_1 \le \frac{v_j}{n_2-1}d_2$. Using that $\frac{\lfloor c_2 \rfloor}{c_2} < d_1$ and $d_2 \le 1$, this implies that $v_j > \lfloor c_2 \rfloor$, which is a contradiction, hence, the considered v does not give an optimal d_2. Similarly, it can be shown that the case where $v_i < \lfloor c_2 \rfloor$ for some i is not optimal.

Thus, for any d_1 it is optimal to take a such that $v_i = \lfloor c_2 \rfloor$ for $p = (n_1 - 1)(\lceil c_2 \rceil - c_2)$ values of i, and $v_i = \lceil c_2 \rceil$ for the remaining i. The value for d_2 as a function of d_1 now easily follows from equality in (5). □

We remark that for fixed a and v, the relation between d_1 and d_2 can be found by considering equality in (5). This relation will be a piece-wise linear function. Further, note that for $c_2 \in \mathbb{N}$ the graph of (4) results in the single point $(1, 1)$, and that setting $d_1 = d_2$ in (4) yields the maximin distance d, with d as in (2). See Figure 4 for a graphical example of the linear function f. This figure shows the set of dominant combinations for $n_1 = 4$ and $n_2 = 8$, including the two extreme dominant combinations $(1, \frac{c_2}{\lceil c_2 \rceil}) = (1, 0.7778)$ and $(\frac{\lfloor c_2 \rfloor}{c_2}, 1) = (0.8571, 1)$. Moreover, the line $d_1 = d_2$ intersects the dominant set exactly in the maximin combination $(d, d) = (0.9130, 0.9130)$.

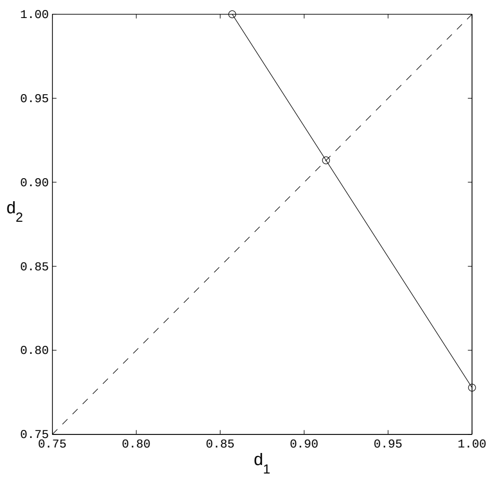

Figure 4: All dominant combinations (d_1, d_2) for $n_1 = 4$ and $n_2 = 8$, and the line $d_1 = d_2$.

3 Three nested sets

We now discuss the case of three nested sets, i.e. $m = 3$. Section 3.1 starts with the general problem formulation. Since we are not able to come up with an explicit formula for the maximin distance we use mixed integer linear programming to solve the problem for several n_1, n_2, n_3. Fortunately, a lower bound on the maximin distance can still be proven. Section 3.2 discusses dominant combinations and in Section 3.3 a heuristic that yields extremely good nested designs is developed.

3.1 Maximin distance

The general problem for three nested sets can be formalized as the following mathematical program:

$$
\begin{array}{lll}
\max\limits_{i=1,2,3;\, j\neq k} & \min\limits_{j,k\in I_i} & (n_i-1)|x_j-x_k| \\
\text{s.t.} & I_1 \subseteq I_2 \subseteq I_3 & \\
 & |I_i| = n_i, & i=1,2 \\
 & 0 \le x_j \le 1, & j \in I_3.
\end{array}
\tag{6}
$$

As in Section 2.1 we may choose without loss of generality $x_1 = 0$, $x_{n_3} = 1$, $x_i < x_{i+1}$, $1 \in I_1$, $n_3 \in I_1$, $1 \in I_2$, and $n_3 \in I_2$. For a given I_2, containing the indices, say, $1 = b_1 < b_2 < \cdots < b_{n_2} = n_3$ we introduce the sequence $w = (w_1, \ldots, w_{n_2-1})$ given by $w_j = b_{j+1} - b_j$. Given an I_1 contained in this I_2 we let $1 = a_1 < a_2 < \cdots < a_{n_1} = n_2$ be such that $b_{a_i} \in I_1$ for $i = 1, \ldots, n_1$. We warn the reader that in this case $\{a_i | i = 1, \ldots, n_1\} \neq I_1$. As before, we let $v_i = a_{i+1} - a_i$. Thus $v_i - 1$ gives the number of additional points of X_2 between the i-th and $(i+1)$-st point of X_1, while $w_j - 1$ gives the number of additional points of X_3 between the j-th and $(j+1)$-st point of X_2. Now the analogue of Lemma 1 is the following.

Lemma 2 *For fixed I_1, I_2, and corresponding a, b, v, w, the optimal value $\delta_{a,w}$ equals*

$$
\Big(\sum_{i=1}^{n_1-1} \max\{ \sum_{j=a_i}^{a_{i+1}-1} \max\{\frac{w_j}{n_3-1}, \frac{1}{n_2-1}\}, \frac{1}{n_1-1}\}\Big)^{-1}.
$$

We would now have to maximize $\delta_{a,w}$ over all appropriate sequences a and w. Unfortunately, we are not able to come up with an explicit formula for the maximin distance, as we did for two nested sets in Section 2.1. However, we can rewrite (6) as a mixed integer linear program:

$$
\begin{array}{lll}
\max & d & \\
\text{s.t.} & d \le (n_3-1)(x_{j+1}-x_j), & j \in I_3 \setminus \{n_3\} \\
 & d \le (n_i-1)(x_k-x_j) + 2 - z_{ik} - z_{ij}, & i=1,2;\ j,k\in I_3;\ j<k \\
 & \sum_{j=1}^{n_3} z_{ij} = n_i, & i = 1,2 \\
 & z_{1j} \le z_{2j}, & j \in I_3 \\
 & 0 \le x_j \le 1, & j \in I_3 \\
 & z_{ij} \in \{0,1\}, & i=1,2;\ j \in I_3.
\end{array}
\tag{7}
$$

Here, $z_{ij} = 1$ if $j \in I_i$, and $z_{ij} = 0$ otherwise. The constraints $\sum_{j=1}^{n_3} z_{ij} = n_i$ and $z_{1j} \le z_{2j}$ insure that $|I_i| = n_i$ and $I_1 \subseteq I_2$, respectively. Using (7) and the *XA Mixed Integer Solver* we found results up to $n_3 = 25$, with computation times varying from 1 second to almost 2.5 hours for some instances, on a PC with a *2000-MHz Pentium IV* processor.

As an example of a nested maximin design, take $n_1 = 4$, $n_2 = 8$, and $n_3 = 18$. Solving (7) for this instance yields the sets $I_1 = \{1, 7, 12, 18\}$ and $I_2 = \{1, 4, 7, 10, 12, 14, 16, 18\}$, implying that $X_1 = \{x_1, x_7, x_{12}, x_{18}\}$ and $X_2 = \{x_1, x_4, x_7, x_{10}, x_{12}, x_{14}, x_{16}, x_{18}\}$, which gives $d = \frac{357}{398} \simeq 0.8970$. See Figure 5 for a graphical representation of the design.

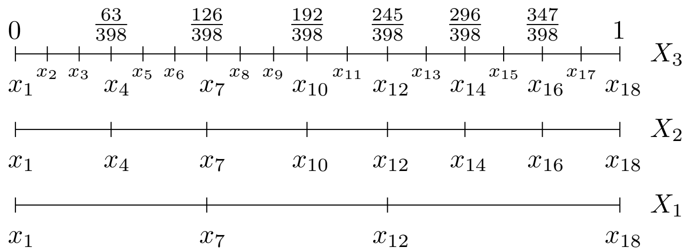

Figure 5: A nested maximin design for $n_1 = 4$, $n_2 = 8$, and $n_3 = 18$, with $d = \frac{357}{398} \simeq 0.8970$.

Although we do not have an explicit formula for the maximin distance, we can prove a general lower bound on this distance. To accomplish this, let $d(n_1, n_2, n_3)$ be the optimal value for d as function of n_1, n_2, n_3, and consider the following lemma.

Lemma 3 *Let $2 \le n_1 \le n_2 \le n_3$. Then $d(n_1, n_2, n_3) \le d(n_1, n_2, n_3 + n_2 - 1)$.*

Proof. Consider any a and w for the problem of (n_1, n_2, n_3). For the problem of $(n_1, n_2, n_3 + n_2 - 1)$ we consider the same a, and w' which is given by $w'_j = w_j + 1$ for all j. Since

$$\max\{\frac{w_j + 1}{n_3 + n_2 - 1 - 1}, \frac{1}{n_2 - 1}\} \le \max\{\frac{w_j}{n_3 - 1}, \frac{1}{n_2 - 1}\},$$

which is easy to show, this implies that $\delta_{a,w'}(n_1, n_2, n_3 + n_2 - 1) \ge \delta_{a,w}(n_1, n_2, n_3)$, and the result follows. □

Proposition 4 *Let $2 \le n_1 \le n_2 \le n_3$. Then $1 \ge d(n_1, n_2, n_3) > (6 - 3\sqrt[3]{4})^{-1} \simeq 0.807887$.*

Proof. Let (again) $c_2 = \frac{n_2 - 1}{n_1 - 1}$ and $c_3 = \frac{n_3 - 1}{n_2 - 1}$. First, note that $d(n_1, n_2, n_3) = 1$ if and only if $c_2, c_3 \in \mathbb{N}$. Because of Lemma 3 we may assume without loss of generality that $c_3 < 2$. To prove the stated inequality, we shall give an a and w such that $\delta_{a,w}(n_1, n_2, n_3) > (6 - 3\sqrt[3]{4})^{-1}$.

Let a be such that the corresponding v takes the value $v_i = \lfloor c_2 \rfloor$ for $i = 1, \dots, p$, with $p = (n_1 - 1)(\lceil c_2 \rceil - c_2)$, and $v_i = \lceil c_2 \rceil$ for the remaining i, i.e. it is the optimal a for two nested sets. Since $c_3 < 2$, it is possible to take w such that w_j is one or two for all j, and we shall do so. To further describe w, we distinguish between two cases.

If $n_3 - n_2 \ge \lfloor c_2 \rfloor (n_1 - 1)(\lceil c_2 \rceil - c_2)$, then we let w be such that $w_j = 2$ for $j = 1, \dots, n_3 - n_2$, and $w_j = 1$ for the remaining j. Since $\max\{\frac{2}{n_3 - 1}, \frac{1}{n_2 - 1}\} = \frac{2}{n_3 - 1}$, we have that

$$\begin{aligned} \delta_{a,w}^{-1} &= (n_1 - 1)(\lceil c_2 \rceil - c_2) \max\{\frac{2\lfloor c_2 \rfloor}{n_3 - 1}, \frac{1}{n_1 - 1}\} + \\ &\quad (n_3 - n_2 - \lfloor c_2 \rfloor (n_1 - 1)(\lceil c_2 \rceil - c_2))\frac{2}{n_3 - 1} + (n_2 - 1 - (n_3 - n_2))\frac{1}{n_2 - 1} \\ &= (\lceil c_2 \rceil - c_2) \max\{0, 1 - \frac{2\lfloor c_2 \rfloor}{c_2 c_3}\} + 4 - c_3 - \frac{2}{c_3}. \end{aligned}$$

Thus, if $2\lfloor c_2 \rfloor < c_2 c_3$, then $\delta_{a,w}^{-1} = (\lceil c_2 \rceil - c_2)(1 - \frac{2\lfloor c_2 \rfloor}{c_2 c_3}) + 4 - c_3 - \frac{2}{c_3}$. Call this expression $f(c_2)$, then it is easy to see that $f(c_2 + 1) < f(c_2)$, hence, we may restrict our attention to the case where $1 < c_2 < 2$. From the above we now obtain that $\delta_{a,w}^{-1} = 6 - c_2 - c_3 - \frac{4}{c_2 c_3}$. This expression is at most $6 - 3\sqrt[3]{4}$, a value that is attained only if $c_2 = c_3 = \sqrt[3]{4}$. The case $2\lfloor c_2 \rfloor \ge c_2 c_3$ is straightforward (then $\delta_{a,w}^{-1} \le 4 - 2\sqrt{2}$), so for the case $n_3 - n_2 \ge \lfloor c_2 \rfloor (n_1 - 1)(\lceil c_2 \rceil - c_2)$ we have proven the lower bound on d.

If $n_3 - n_2 < \lfloor c_2 \rfloor (n_1 - 1)(\lceil c_2 \rceil - c_2)$, then we may assume that c_2 is not an integer. Let $p = (n_1 - 1)(\lceil c_2 \rceil - c_2)$, and introduce $t = \frac{n_3 - n_2}{p} = \frac{c_2(c_3 - 1)}{\lceil c_2 \rceil - c_2}$. It follows that $\lceil t \rceil \le \lfloor c_2 \rfloor$. We now take w as follows: for $m(t - \lfloor t \rfloor)$ values of $i = 1, \dots, m$ we have $\lceil t \rceil$ values of j, $a_i \le j < a_{i+1}$ for which $w_j = 2$, and the remaining $\lfloor c_2 \rfloor - \lceil t \rceil$ of such j-s have $w_j = 1$; for the other values of $i = 1, \dots, m$ we have $\lfloor t \rfloor$ values of j, $a_i \le j < a_{i+1}$ for which $w_j = 2$, and the remaining $\lfloor c_2 \rfloor - \lfloor t \rfloor$ of such j-s have $w_j = 1$; and for all $j \ge a_{m+1}$ we have $w_j = 1$. From Lemma 2 we now find that

$$\begin{aligned} \delta_{a,w}^{-1} &= m(t - \lfloor t \rfloor) \max\{\lceil t \rceil \frac{2}{n_3 - 1} + (\lfloor c_2 \rfloor - \lceil t \rceil)\frac{1}{n_2 - 1}, \frac{1}{n_1 - 1}\} + \\ &\quad m(1 - t + \lfloor t \rfloor) \max\{\lfloor t \rfloor \frac{2}{n_3 - 1} + (\lfloor c_2 \rfloor - \lfloor t \rfloor)\frac{1}{n_2 - 1}, \frac{1}{n_1 - 1}\} + \\ &\quad (n_1 - 1)(c_2 - \lfloor c_2 \rfloor)\lceil c_2 \rceil \frac{1}{n_2 - 1} \\ &= \frac{\lceil c_2 \rceil - c_2}{c_2}(t - \lfloor t \rfloor) \max\{\lceil t \rceil (\frac{2}{c_3} - 1), c_2 - \lfloor c_2 \rfloor\} + \\ &\quad \frac{\lceil c_2 \rceil - c_2}{c_2}(1 - t + \lfloor t \rfloor) \max\{\lfloor t \rfloor (\frac{2}{c_3} - 1), c_2 - \lfloor c_2 \rfloor\} + 1. \end{aligned}$$

We now assume that $\lfloor t \rfloor (\frac{2}{c_3} - 1) < c_2 - \lfloor c_2 \rfloor < \lceil t \rceil (\frac{2}{c_3} - 1)$. In the other cases it is straightforward to show that $\delta_{a,w}^{-1} \le 4 - 2\sqrt{2}$. Then

$$\delta_{a,w}^{-1} = \frac{\lceil c_2 \rceil - c_2}{c_2}(t - \lfloor t \rfloor)\lceil t \rceil (\frac{2}{c_3} - 1) + \frac{\lceil c_2 \rceil - c_2}{c_2}(1 - t + \lfloor t \rfloor)(c_2 - \lfloor c_2 \rfloor) + 1. \tag{8}$$

Using the left inequality in above assumption we find that

$$\begin{aligned}\delta_{a,w}^{-1} &< \frac{\lceil c_2\rceil - c_2}{c_2}(t-\lfloor t\rfloor)(\frac{2}{c_3}-1)+\frac{\lceil c_2\rceil - c_2}{c_2}(c_2-\lfloor c_2\rfloor)+1 \\ &\leq (c_3-1)(\frac{2}{c_3}-1)+\frac{\lceil c_2\rceil - c_2}{c_2}(c_2-\lfloor c_2\rfloor)+1 \\ &\leq 4-2\sqrt{2}+\frac{\lceil c_2\rceil - c_2}{c_2}(c_2-\lfloor c_2\rfloor).\end{aligned}$$

If $c_2 > 4$, then this upper bound suffices (its maximum is attained at $c_2 = \sqrt{20}$), as one can easily check. For $c_2 < 4$, fix $k = \lceil t\rceil$ (≤ 3), and let $c_2 > k$. Then (8) reduces to

$$\delta_{a,w}^{-1} = 1+3k-kc_3-\frac{2k}{c_3}-(c_3-1)(c_2-\lfloor c_2\rfloor)+k\frac{\lceil c_2\rceil - c_2}{c_2}(c_2-\lfloor c_2\rfloor-(k-1)(\frac{2}{c_3}-1)),$$

the maximum of which is attained for some c_2 between k and $k+1$, i.e. $\lfloor c_2\rfloor$ is minimal. For each $k = 1, 2, 3$ (separately) it is now possible to obtain an appropriate upper bound on $\delta_{a,w}^{-1}$, under the assumptions that $k \leq c_2 \leq k+1$ and $1 \leq c_3 \leq 2$. For $k = 1$, this upper bound is $6-3\sqrt[3]{4}$, and it is attained when $c_2 = c_3 = \sqrt[3]{2}$. □

Note that the obtained lower bound is tight since we can take c_2 and c_3 arbitrarily close to $\sqrt[3]{2}$, and in these cases the given a and w are optimal; see Proposition 6. The interpretation of this lower bound is that for all values of n_1, n_2, n_3, by nesting the sets X_1, X_2, X_3 we will never lose more than 19.21%, with respect to the "restriction free" maximin distance. In practice this implies that a linking parameter can be included in the maximin designs, at a cost of using designs that are at most 19.21% worse with respect to space-fillingness.

Applying our approach in case of (three-stage) sequential evaluations incurs a loss of $1-d$ when one stage suffices. If two stages are sufficient we obtain a net gain of $d-d'$, where $d' = \frac{c_2}{\lceil c_2\rceil}$ (see Section 2.1), and when all three stages are needed we gain $d-d''$, where $d'' \leq d' \leq d$ and $d'' > \frac{1}{2}$ for $c_3 > 1$. Thus, the net gain of our approach equals $(d-1)+(d-d')+(d-d'')$ and it takes values in the interval $[-0.19, 0.84]$ for $n_3 \leq 25$.

3.2 Dominance

The notion of dominance was introduced in Section 2.2. Similar as before, we will call a combination (d_1, d_2, d_3) dominant if it is not possible to improve one of the coordinates, without deteriorating another coordinate. Unlike with two nested sets, the maximin combination (d, d, d) is not necessarily dominant, e.g. $d(4, 8, 17) = 0.9130$, however, $(0.9130, 0.9130, 0.9275)$ is dominant. In Section 2.2 we showed that $(1, \frac{c_2}{\lceil c_2\rceil})$ and $(\frac{\lfloor c_2\rfloor}{c_2}, 1)$ are extreme dominant combinations for two nested sets. Extending these ideas to three nested sets, i.e. fixing $d_i = 1$ and maximizing d_j, $j \neq i$, leads to extreme dominant combinations. Note that the extreme dominant combinations are again lower bounds on the maximin distance $d = d(n_1, n_2, n_3)$. An upper bound on d is obtained by the simple observation that $d(n_1, n_2, n_3) \leq \max\{d(n_1, n_2), d(n_1, n_3), d(n_2, n_3)\}$. Furthermore, it is easily shown that $d(n_1, n_2, n_3) = d(n_2, n_3)$ if and only if $c_2 \in \mathbb{N}$, and $d(n_1, n_2, n_3) = d(n_1, n_2)$ if and only if $c_3 \in \mathbb{N}$.

All this may lead to the believe that we can extend the idea of finding the maximin distance by means of extreme dominant combinations, like we did for two nested sets. As an example, from Figure 6 it can be seen that the dominant combinations for $n_1 = 4$, $n_2 = 8$, and $n_3 = 18$, lie in a plane through the extreme dominant combinations $(1, 0.7778, 0.9444)$, $(0.8571, 1, 0.8095)$, and $(0.8824, 0.8235, 1)$. This plane intersects the line $d_3 = d_2 = d_1$ exactly in the maximin combination $(0.8970, 0.8970, 0.8970)$, strengthening the believe that this method also works for three nested sets. Unfortunately, the dominant combinations will not always fall in a plane through the extreme dominant points; see Figure 7 for an example of this. Furthermore, this plane can not always be used to find the maximin combination. For example, take $n_1 = 6$, $n_2 = 8$, and $n_3 = 12$. Then the plane through the extreme dominant combinations $(1, 0.7, 0.7333)$, $(0.7143, 1, 0.7857)$, and $(0.9091, 0.6364, 1)$, results in the unattainable combination $(0.8324, 0.8324, 0.8324)$, when intersected with the line $d_3 = d_2 = d_1$, thereby "missing" the correct maximin combination $(0.8262, 0.8262, 0.8262)$.

3.3 Heuristic

In the previous section we showed that, when dealing with three nested sets, the maximin distance cannot always be found by means of extreme dominant combinations. Note that even when this method would

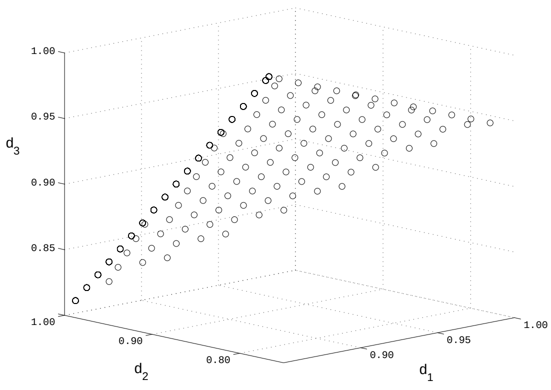

Figure 6: Dominant combinations for $n_1 = 4$, $n_2 = 8$, and $n_3 = 18$.

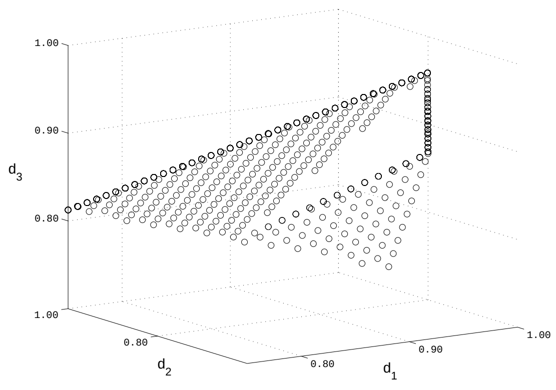

Figure 7: Dominant combinations for $n_1 = 4$, $n_2 = 9$, and $n_3 = 14$.

work, we still had to find a way to construct the corresponding nested maximin designs. Mixed integer programming can be used; however, it was found to be too slow in finding nested maximin designs for large values of n_1, n_2, n_3. To deal with these problems we built a heuristic that searches for a good nested design, and, hence, a good distance.

Our heuristic is based on the observation that all nested maximin designs that were found by solving (7) contained the corresponding two nested sets assignments, as given in (3), as part of their solutions, e.g. compare Figure 2 and 5. Therefore, for given n_1, n_2, n_3 ($c_2, c_3 \notin \mathbb{N}$), we first use (3) to construct a nested maximin design on n_1, n_2. Every interval $[x_l, x_{l+1}]$, $l \in I_2 \setminus \{n_2\}$, then will have a width of at least $\frac{d}{n_2-1}$, where $d = d(n_1, n_2)$ is as in (2). This implies that we can add up to q points to each interval without decreasing d, as long as q fulfills the inequality

$$\frac{d(q+1)}{n_3-1} \leq \frac{d}{n_2-1}, \text{ or equivalently } q+1 \leq c_3,$$

which results in at most $q = \lfloor c_3 \rfloor - 1$ additional points per interval, or $(\lfloor c_3 \rfloor - 1)(n_2 - 1)$ in total. Hence, if $n_3 - n_2 \leq (\lfloor c_3 \rfloor - 1)(n_2 - 1)$, we are finished, since spreading the $n_3 - n_2$ points equally over the $n_2 - 1$ intervals will yield a nested maximin design with distance $d(n_1, n_2, n_3) = d(n_1, n_2)$.

If $n_3 - n_2 > (\lfloor c_3 \rfloor - 1)(n_2 - 1)$, we add q points to every interval and have $r = (n_3 - n_2) - (\lfloor c_3 \rfloor - 1)(n_2 - 1) < (n_3 - n_2) - (c_3 - 2)(n_2 - 1) = n_2 - 1$ points remaining. These remaining r points are then sequentially added to one of the $n_2 - 1$ intervals as follows. Consider the case where s points are already assigned, $s \in \{0, \ldots, r-1\}$, and consider the index sets $I_1^s \subseteq I_2^s \subseteq I_3^s = \{1, \ldots, n_3'\}$, which describe the current nested design on n_1, n_2, n_3', where $n_3' = n_2 + (\lfloor c_3 \rfloor - 1)(n_2 - 1) + s$. Then the corresponding maximal distance can readily be computed using Lemma 2.

When assigning the $(s+1)$-st point we first compute for each of the $n_2 - 1$ intervals what the maximal distance will be if the point is assigned to that particular interval, again using Lemma 2. Naturally, the interval for which this distance is the largest is chosen and the corresponding $I_1^{s+1} \subseteq I_2^{s+1} \subseteq I_3^{s+1}$ describe the new nested design. In case of a tie we choose that interval for which $\max\{\frac{(I_2^{s+1})_{i+1} - (I_2^{s+1})_i}{n_3-1}, \frac{1}{n_2-1}\} - \max\{\frac{(I_2^s)_{i+1} - (I_2^s)_i}{n_3-1}, \frac{1}{n_2-1}\}$ is the smallest, $i = 1, \ldots, n_2 - 1$; see (3), where $(I_2^s)_i$ is the i-th element of set I_2^s. This value can be seen as the relative cost of adding an extra point to a particular interval. Leaving out this second objective may result in bad nested designs.

For given index sets I_1, I_2, I_3 it takes $\mathcal{O}(n_1 n_2)$ time to compute the maximal distance, using Lemma 2. There are $s \leq r < n_2$ extra points to be added and for each of these points $n_2 - 1$ index sets have to be considered, hence, we have to apply Lemma 2 $\mathcal{O}({n_2}^2)$ times. Therefore, a nested design for n_1, n_2, n_3 is found in $\mathcal{O}(n_1 {n_2}^3)$ time. Note that the complexity does not depend on n_3. Moreover, it turns out that our heuristic yields an optimal nested design for all values of n_1, n_2, n_3 we solved so far, i.e. for $n_3 \leq 25$. We conjecture that the heuristic will find a nested maximin design for all n_1, n_2, n_3.

4 Four or more nested sets

In this section we discuss the case of $m \geq 4$ nested sets. This can be formalized as the following mathematical program:

$$\begin{array}{lll} \max & \min\limits_{\substack{j,k\in I_i;\, j\neq k \\ i=1,\ldots,m}} (n_i-1)|x_j-x_k| & \\ \text{s.t.} & I_1 \subseteq I_2 \subseteq \cdots \subseteq I_m & \\ & |I_i| = n_i, & i=1,\ldots,m-1 \\ & 0 \leq x_j \leq 1, & j \in I_m. \end{array} \tag{9}$$

Furthermore, Lemmas 2 and 3 can easily be generalized to more nested sets. In particular,

Lemma 4 *Let $2 \leq n_1 \leq \cdots \leq n_m$. Then $d(n_1,\ldots,n_{m-1},n_m) \leq d(n_1,\ldots,n_{m-1},n_m+n_{m-1}-1)$.*

Now, we consider the case $n_m < 2n_1$. Let $c_i = \frac{n_i-1}{n_{i-1}-1}$, $i=2,\ldots,m$, and $d = d(n_1,\ldots,n_m)$. For fixed I_1, let it contain the indices $1 = a_1 < a_2 < \cdots < a_{n_1} = n_m$. Note that this a is somewhat different from a in the previous section, in the sense that it here gives the relation between I_1 and I_m, instead of between I_1 and I_2. As before, let the sequence $v = (v_1,\ldots,v_{n_1-1})$ be given by $v_i = a_{i+1}-a_i$. Thus $v_i - 1$ gives the number of additional points of X_m between the i-th and $(i+1)$-st point of X_1.

Proposition 5 *Let $m \geq 3$ and $2 \leq n_1 \leq \cdots \leq n_m < 2n_1$. Then the maximal value for d equals*

$$\frac{1}{2m - \frac{2}{c_2} - \cdots - \frac{2}{c_m} - c_2c_3\cdots c_m}.$$

Proof. Consider an I_1 such that the corresponding v takes only values 1 and 2, i.e. between two neighboring points from X_1 there is at most one point from X_m. Since $\max\{\frac{2}{n_j-1},\frac{1}{n_1-1}\} = \frac{2}{n_j-1}$, and since the number of i such that $v_i = 1$ equals $2n_1-n_m-1$, it follows that $\delta_v = ((2n_1-n_m-1)\frac{1}{n_1-1} + \sum_{j=2}^m (n_j - n_{j-1})\frac{2}{n_j-1})^{-1} = (2m - 2c_2^{-1} - \cdots - 2c_m^{-1} - c_2c_3\cdots c_m)^{-1}$. That this v gives the optimal d can be shown by comparing $\delta_{v'}^{-1}$, for a v' with $v'_i \geq 3$ for some i, to $\delta_{v''}^{-1}$, where v'' is obtained from v' by letting $v''_i = v'_i - 1$, and taking $v''_j = 2$ for a j with $v'_j = 1$. Such a j exists because of the condition $n_m < 2n_1$. We omit further technical details. □

Using Proposition 5, it is easy to show that the following holds:

Proposition 6 *Let $m \geq 2$ and $2 \leq n_1 \leq \cdots \leq n_m < 2n_1$. Then $1 \geq d > (2m(1-\sqrt[m]{\tfrac{1}{2}}))^{-1}$.*

The lower bound for d is attained when $c_i = \sqrt[m]{2}$ for all i. We conjecture that this lower bound for d holds in all cases. This conjecture is supported by the results for $m = 2$ and $m = 3$, see Proposition 2 and Proposition 4, respectively.

We remark further that the sequence $(2m(1-\sqrt[m]{\tfrac{1}{2}}))^{-1}$ is decreasing in m, and converges to $\frac{1}{2\log 2} \simeq 0.721348$. Hence, if our conjecture is true we will never lose more than 27.87%, with respect to the "restriction free" maximin distance, when nesting the sets $X_1,\ldots,X_m$.

For the case $m = 4$ we extended the mixed integer linear program for three nested sets, see (7), and found results up to $n_4 = 19$. Unfortunately, as n_4 gradually increases, the computation time rapidly grows, leading to some instances that take 4 hours to solve. Therefore, we built a heuristic that searches for good nested designs for given n_1, n_2, n_3, n_4. This heuristic first constructs a nested design for n_1, n_2, n_3, see Section 3.3, which is conjectured to be an optimal nested design. Then the $n_4 - n_3$ extra points are sequentially added, in the way described in Section 3.3. As can be observed from Figure 8, for $n_4 \leq 19$ our heuristic often finds the maximin distance (and thus the corresponding nested maximin design), and is not too far off in most other cases. Unfortunately, there is an instance, i.e. $(4,6,9,14)$, for which the maximal distance found by the heuristic has a value that is less than the (conjectured) lower bound in Proposition 6 (the dotted lines in the figure). For this instance the heuristic finds a maximal distance of 0.7796, which is smaller than the lower bound of 0.7857 and the maximin distance $d(4,6,9,14) = 0.7923$.

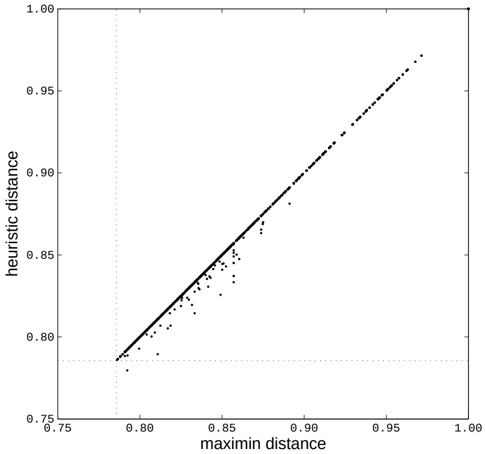

Figure 8: Maximal distance found by heuristic versus the maximin distance for $n_4 \leq 19$.

5 Conclusions and further research

In this paper we discussed the construction of nested maximin designs. Such designs play an important role in the design of computer experiments in black box evaluation and optimization processes. The two main reasons for using nested maximin designs are linking parameters and sequential evaluations. Linking parameters occur when several black box functions share the same input parameters, or when uniformity in parameter settings is needed. We speak of sequential evaluations when an initial set of function evaluations is followed by extra sets of evaluations, as is often the case in practice.

Constructing a maximin design for two black box functions that share a single linking parameter, or for two-stage sequential evaluations, can be considered as constructing a nested maximin design for two nested sets $X_1 \subseteq X_2 = \{x_1, \ldots, x_{n_2}\}$, with $x_j \in [0,1]$. In this case, the maximin distance equals $d = (1 + \lfloor c_2 \rfloor + \lceil c_2 \rceil - c_2 - \lfloor c_2 \rfloor \lceil c_2 \rceil \frac{1}{c_2})^{-1}$, where $c_2 = \frac{n_2-1}{n_1-1}$, and a corresponding nested maximin design is given by (3). It is shown that due to the restriction $X_1 \subseteq X_2$ the resulting designs are at most 14.64% less space-filling than standard maximin designs, in case of linking parameters. For sequential evaluations Figure 3 shows the net gain of using nested maximin designs. In both cases, it turns out that using nested maximin designs instead of traditional maximin designs is very profitable.

Although we lack an explicit formula for the maximin distance of three nested sets it is proven that this distance is at least 0.807887. Furthermore, for small instances nested maximin designs can be found by mixed integer programming. Fortunately, we developed a fast heuristic that constructs nested designs for larger instances, too. Based on the results, it is conjectured that this heuristic is optimal, i.e. it will yield a nested maximin design for all instances. An extension of the heuristic to four nested sets often finds nested maximin designs and is not too far off in most other cases.

To investigate the trade-off between nested sets dominant combinations are introduced. In case of two nested sets this relation is linear and is given by (4). For three nested sets the behavior of these dominant combinations is not always that simple, e.g. see Figure 7. Finally, it is proven that a lower bound on the maximin distance for the general problem of m nested sets is given by $(2m(1 - \sqrt[m]{\frac{1}{2}}))^{-1}$, under the restriction $n_m < 2n_1$. It is conjectured that this lower bound also holds for all other instances, as is supported by the results for $m = 2$ and $m = 3$.

Besides considering the maximin criterion, the concept of nested designs could also be applied to other distance measures, like minimax, IMSE, and maximum entropy. Furthermore, the one-dimensional results in this paper could be extended to more dimensions, i.e. constructing multi-dimensional nested maximin designs, which is subject of current research.

References

[1] Booker, A.J., Dennis, J.E., Frank, P.D., Serafini, D.B., Torczon, V., and Trosset, M.W. (1999). A Rigorous Framework for Optimization of Expensive Functions by Surrogates, *Structural Optimization* 17, 1-13.

[2] Hertog, D. den, and Stehouwer, H.P. (2002). Optimizing Color Picture Tubes by High-cost Nonlinear Programming, *European Journal on Operations Research* 140(2), 197-211.

[3] Husslage, B.G.M., Dam, E.R. van, Hertog, D. den, Stehouwer, H.P., and Stinstra, E.D. (2003). Collaborative Metamodeling: Coordinating Simulation-based Product Design, *Concurrent Engineering: Research and Applications* 11(4), 267-278.

[4] Jones, D., Schonlau, M., and Welch, W. (1998). Efficient Global Optimization of Expensive Black-box Functions, *Journal of Global Optimization* 13, 455-492.

[5] Montgomery D.C. (1984). *Design and Analysis of Experiments*, Second Edition, John Wiley & Sons, New York.

[6] Morris, M.D., and Mitchell,T.J. (1995). Exploratory Designs for Computer Experiments, *Journal of Statistical Planning and Inference* 43, 381-402.

[7] Myers, R.H. (1999). Response Surface Methodology – Current Status and Future Directions, *Journal of Quality Technology* 31, 30-74.

[8] Sacks, J., Welch, W.J., Mitchell, T.J., and Wynn, H.P. (1989). Design and Analysis of Computer Experiments, *Statistical Science* 4, 409-435.

[9] Santner, Th. J., Williams, B.J., and Notz, W.I. (2003). *The Design and Analysis of Computer Experiments*, *Springer Series in Statistics*, Springer-Verlag, New York.