MATHEMATICS OF COMPUTATION
Volume 75, Number 254, Pages 859–878
S 0025-5718(05)01806-5
Article electronically published on December 27, 2005

LOWER BOUNDS AND STOCHASTIC OPTIMIZATION ALGORITHMS FOR UNIFORM DESIGNS WITH THREE OR FOUR LEVELS

KAI-TAI FANG, DIETMAR MARINGER, YU TANG, AND PETER WINKER

Abstract. New lower bounds for three- and four-level designs under the centered L_2-discrepancy are provided. We describe necessary conditions for the existence of a uniform design meeting these lower bounds. We consider several modifications of two stochastic optimization algorithms for the problem of finding uniform or close to uniform designs under the centered L_2-discrepancy. Besides the threshold accepting algorithm, we introduce an algorithm named balance-pursuit heuristic. This algorithm uses some combinatorial properties of inner structures required for a uniform design. Using the best specifications of these algorithms we obtain many designs whose discrepancy is lower than those obtained in previous works, as well as many new low-discrepancy designs with fairly large scale. Moreover, some of these designs meet the lower bound, i.e., are uniform designs.

1. Introduction

Proposed in [7, 13], the uniform design has been widely applied in manufacturing, system engineering, pharmaceutics and natural sciences. Many construction methods for uniform designs have been proposed, including the *good lattice point method* ([7, 13]). In general, the previous methods only provide an approximation to the uniform design. The quality of this approximation is often considered as unsatisfactory. More recently, two new construction methods have been applied, i.e., construction via combinatorial design and construction by stochastic optimization. The first method utilizes the properties of various combinatorial configurations as well as construction techniques frequently used in design theory to obtain uniform designs without any computation (see, for example, [2, 3, 10, 11]). Thus, this method can be used to obtain certain infinite classes of uniform designs. But due to the nonexistence of combinatorial configurations, there will always be many constraints to the parameters of the designs. When these constraints are not satisfied, the construction will fail. In this sense, the stochastic optimization approach appears advantageous. Given arbitrary parameters, we can always generate a low-discrepancy design by implementing an effective algorithm. Up to now, the threshold accepting algorithm is most frequently used to construct uniform

Received by the editor November 3, 2004.
2000 *Mathematics Subject Classification.* Primary 68Q17, 68Q15, 62K99.
Key words and phrases. Discrepancy, lower bound, uniform designs, stochastic optimization, threshold accepting.
The work was partially supported by the Grants GER/JRS/03-04/01, RGC/HKBU 200804, FRG/03-04/II-711, and DAAD D/03/314145.

designs (see, for example, [4, 5, 6, 15]). However, due to the high computational complexity of the uniform design problem and the resulting computational burden, this stochastic optimization approach has been limited to finding uniform designs with fairly small scale. In this paper, we present a new combinatorial optimization approach named balanced-pursuit heuristic, which combines knowledge about certain combinatorial properties of low-discrepancy designs with a general heuristic optimization framework. We compare several modifications of this algorithm and the threshold accepting heuristic in order to identify optimal specifications of the algorithms. Finally, we assess the relative performance of both approaches for different settings and provide new low-discrepancy and uniform designs for the three- and four-level case.

The uniform design is a type of "space filling" design for computer experiments ([1]), but if we restrict the design to certain lattice points, the uniform design can also be utilized as a fractional factorial design. Up to now, most of the existing uniform designs are based on the *U-type designs*. A U-type design $U(n; q^m)$ of n runs and m factors with q levels corresponds to an $n \times m$ matrix $X = (x_1, \ldots, x_m)$ such that each column x_i takes values from a set of q elements, say $\{1, 2, \ldots, q\}$, equally often. We use $\mathcal{U}(n; q^m)$ to denote the set of all $U(n; q^m)$ designs. Applying a map $f\colon l \to \frac{2l-1}{2q}$, $l = 1, \ldots, q$, to the n runs of a design $U(n; q^m)$, we can obtain n points in the canonical experimental domain $C^m = [0, 1)^m$. The transferred design will be denoted by $\tilde{U}(n; q^m)$, and the set of all such designs will be denoted by $\tilde{\mathcal{U}}(n; q^m)$. The correspondence f between $\mathcal{U}(n; q^m)$ and $\tilde{\mathcal{U}}(n; q^m)$ is obviously one-to-one and linear.

For uniform designs, the measure of uniformity is important. Historically, the star discrepancy has been used first, e.g., in quasi-Monte Carlo methods (or number-theoretic methods) and in uniform design theory (see [7] and [12]). The L_p-discrepancy provides a generalization of this concept. However, [8, 9] pointed out that the L_p-discrepancy exhibits some weakness and further proposed several modifications of the L_p-discrepancy. Among those, the centered L_2-discrepancy (CD_2) is the most important and attractive one.

Let $\mathcal{P}$ be a set of n points in C^m. Then, the centered L_2-discrepancy is defined as follows:

$$(CD_2(\mathcal{P}))^2 = \sum_{u \neq \emptyset} \int_{C^u} \left[\frac{N(\mathcal{P} \cap J_w(\boldsymbol{x}_u))}{n} - \mathrm{Vol}(J_w(\boldsymbol{x}_u)) \right]^2 d\boldsymbol{x}_u,$$

where u is any nonempty subset of the coordinate indices $\{1, 2, \ldots, m\}$, $|u|$ denotes the cardinality of u, C^u is the $|u|$-dimensional unit cube involving the coordinates in u, $N(\mathcal{P} \cap A)$ denotes the number of points of $\mathcal{P}$ falling in A, $\boldsymbol{x}_u$ is the projection of $\boldsymbol{x} \in \mathcal{P}$ onto C^u, and $J_w(\boldsymbol{x}_u)$ is the hyper-rectangle in C^u containing the points between $\boldsymbol{x}_u$ and the nearest vertex of C^u.

An analytical expression for $CD_2(\mathcal{P})$ is given by [8]. Using this result and the definitions

$$\begin{aligned} \alpha_i^l &= 1 + \frac{1}{2} \cdot \left| x_{il} - \frac{1}{2} \right| - \frac{1}{2} \cdot \left| x_{il} - \frac{1}{2} \right|^2, \\ \beta_{ij}^l &= 1 + \frac{1}{2} \cdot \left| x_{il} - \frac{1}{2} \right| + \frac{1}{2} \cdot \left| x_{jl} - \frac{1}{2} \right| - \frac{1}{2} \cdot |x_{il} - x_{jl}|, \end{aligned}$$

where $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{im}) \in \mathcal{P}$,[1] we obtain (see also [5])

$$
\begin{aligned}
&(CD_2(\mathcal{P}))^2 \\
&\quad = \left(\frac{13}{12}\right)^m - \frac{2}{n} \cdot \sum_{i=1}^{n} \prod_{l=1}^{m} \alpha_i^l + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \prod_{l=1}^{m} \beta_{ij}^l \\
&\quad = \left(\frac{13}{12}\right)^m - \frac{2}{n} \cdot \sum_{i=1}^{n} \prod_{l=1}^{m} \alpha_i^l + \frac{1}{n^2} \cdot \sum_{i=1}^{n} \prod_{l=1}^{m} \beta_{ii}^l + \frac{1}{n^2} \cdot \sum_{i=1}^{n} \sum_{j \neq i} \prod_{l=1}^{m} \beta_{ij}^l \\
&\quad = \left(\frac{13}{12}\right)^m + \frac{1}{n^2} \cdot \left(- \sum_{i=1}^{n} 2 \cdot n \cdot \prod_{l=1}^{m} \alpha_i^l + \sum_{i=1}^{n} \prod_{l=1}^{m} \beta_{ii}^l \right) \\
&\qquad + \frac{1}{n^2} \cdot \sum_{i=1}^{n} \sum_{j \neq i} \prod_{l=1}^{m} \beta_{ij}^l \,.
\end{aligned} \tag{1.1}
$$

Furthermore, let

$$
\sigma_i^{(1)} = 2 \cdot n \cdot \prod_{l=1}^{m} \alpha_i^l, \qquad \sigma_i^{(2)} = \prod_{l=1}^{m} \beta_{ii}^l \left(= \prod_{l=1}^{m} \alpha_i^l \right), \qquad \sigma_{ij} = \prod_{l=1}^{m} \beta_{ij}^l. \tag{1.2}
$$

Then, (1.1) can be rewritten as

$$
(CD_2(\mathcal{P}))^2 = \left(\frac{13}{12}\right)^m + \frac{1}{n^2} \cdot \sum_{i=1}^{n} \underbrace{\left(-\sigma_i^{(1)} + \sigma_i^{(2)} \right)}_{=\sigma_i} + \frac{1}{n^2} \cdot \sum_{i \neq j} \sigma_{ij} \,. \tag{1.3}
$$

In passing, note that σ_i provides the negative of a measure of the distance of x_i from the central point $(\frac{1}{2}, \ldots, \frac{1}{2})$, while the σ_{ij} provide information on pairwise distances between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$.

Given the closed form expression for the centered L_2-discrepancy in (1.3), the application of stochastic optimization algorithms appears feasible. However, we provide two additional results which increase the efficiency of any local search optimization heuristic in this context. First, from (1.3) it is possible to obtain a formula for a local update of the objective function if only some elements of the design are modified. Second, for the cases of three- and four-level designs, we provide lower bounds for their discrepancy. Thus, it becomes possible to prove that a low-discrepancy design is a uniform design in its proper sense, i.e., has minimal discrepancy.

The paper is organized as follows. Focused on the special cases for levels three and four, Section 2 presents lower bounds for the centered L_2-discrepancy on the set of U-type designs, and also analyzes the combinatorial properties of a U-type design achieving these lower bounds. Based on these combinatorial properties, Section 3 introduces the balance-pursuit heuristic in detail and discusses several modifications of this new algorithm and the threshold accepting heuristic. In particular, we study several neighborhood structures on the search space of U-type designs. Section 4 provides a comparison of the performance of the different approaches and computational results. The final section summarizes our main findings and suggests possible extensions for future research.

[1] Throughout our paper, the set of points $\mathcal{P}$ is always chosen from a transferred U-type design $\tilde{U}(n; q^m)$.

2. Special cases for three-level and four-level designs

Let us first concentrate on the analytical expression (1.1) for three-level designs. In this case, the variable $\alpha_i^l = 1 + \frac{1}{2}|x_{il} - \frac{1}{2}| - \frac{1}{2}|x_{il} - \frac{1}{2}|^2$ can only take two possible values, i.e., 1 or $\frac{10}{9}$, and the variable $\beta_{ij}^l = 1 + \frac{1}{2}|x_{il} - \frac{1}{2}| + \frac{1}{2}|x_{jl} - \frac{1}{2}| - \frac{1}{2}|x_{il} - x_{jl}|$ can also take two possible values, i.e., 1 or $\frac{4}{3}$. Thus, for three-level designs the formula for $(CD_2(\mathcal{P}))^2$ can be simplified to

$$(CD_2(\mathcal{P}))^2 = \left(\frac{13}{12}\right)^m - \frac{2}{n}\sum_{i=1}^{n}\left(\frac{10}{9}\right)^{\gamma_i} + \frac{1}{n^2}\sum_{i=1}^{n}\left(\frac{4}{3}\right)^{\gamma_i} + \frac{1}{n^2}\sum_{i\neq j}\left(\frac{4}{3}\right)^{\gamma_{ij}}, \tag{2.1}$$

where

$$\gamma_i = \sharp\{x_{il} \neq \frac{1}{2}, l = 1, \ldots, m\}$$

and

$$\gamma_{ij} = \sharp\{(x_{il}, x_{jl}) : x_{il} = x_{jl} \neq \frac{1}{2}, l = 1, \ldots, m\}.$$

For a further analysis of the above expression, we first discuss the property of the function $f(x) = \frac{1}{3}(\frac{4}{3})^x - \frac{2n}{9}(\frac{10}{9})^x$. Obviously, the function $f(x)$ has a single root, i.e.,

$$x_0 = \frac{\log\left(\frac{2n}{3}\right)}{\log\left(\frac{6}{5}\right)},$$

and the first derivative of $f(x)$,

$$\frac{df(x)}{dx} = \frac{1}{3}\left(\frac{4}{3}\right)^x \log\left(\frac{4}{3}\right) - \frac{2n}{9}\left(\frac{10}{9}\right)^x \log\left(\frac{10}{9}\right),$$

also has exactly one root point, i.e.,

$$x_1 = \frac{\log\left(\frac{2n}{3}\right) + \log\log\left(\frac{10}{9}\right) - \log\log\left(\frac{4}{3}\right)}{\log\left(\frac{6}{5}\right)}.$$

So $f(x)$ is strictly decreasing on $(-\infty, x_1]$, and strictly increasing on $[x_1, \infty)$. Moreover, it is easy to see that $\lim_{x\to-\infty} f(x) = 0$ and $\lim_{x\to\infty} f(x) = \infty$. For the case $x_1 \leq 0$, we will have $f(y_2) < f(y_1)$ for any $0 \leq y_2 < y_1$, while for the case $0 < x_1$, we can plot a graph of $f(x)$ as provided in Figure 1.

Note the point M with coordinates $(x_\mu, f(x_\mu))$ indicated in Figure 1. The function $f(x)$ has the same value at M as at $x = 0$, i.e., $f(x_\mu) = f(0)$. Consequently, for any $y_1 \geq x_\mu$ and $0 \leq y_2 < y_1$, we will always have $f(y_1) > f(y_2)$. Now, we are in a position to prove the following two theorems.

Theorem 1. *For a U-type design $U(n, 3^m)$, if $\mu = \frac{2}{3}m$ and $\gamma = \frac{2m(n-3)}{9(n-1)}$ are both integers and $f(\mu) \geq f(0)$, then*

$$(CD_2(\mathcal{P}))^2 \geq \left(\frac{13}{12}\right)^m - 2\left(\frac{10}{9}\right)^\mu + \frac{1}{n}\left(\frac{4}{3}\right)^\mu + \frac{n-1}{n}\left(\frac{4}{3}\right)^\gamma. \tag{2.2}$$

The above lower bound can be obtained if and only if $\gamma_i = \mu$, $\gamma_{ij} = \gamma$, for all $i \neq j$, where γ_i and γ_{ij} appear in (2.1).

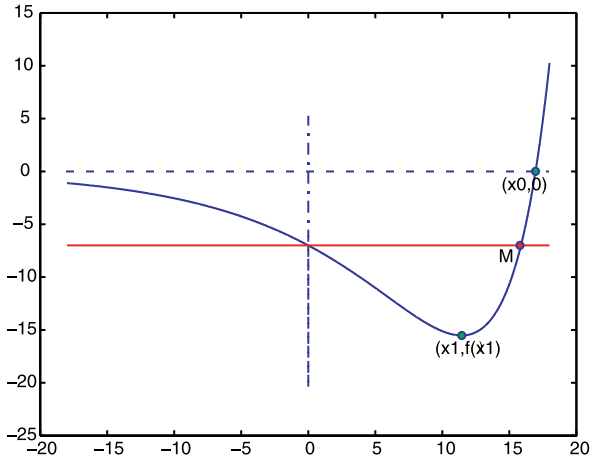

FIGURE 1. Graph of $f(x)$ in the case of $0 < x_1$

Proof. Since $\sum_{i\neq j}\gamma_{ij} = \frac{2mn(n-3)}{9}$ is a constant for any U-type design $U(n, 3^m)$, from the arithmetic and geometric mean inequality, it is straightforward to show that $\frac{1}{n^2}\sum_{i\neq j}(\frac{4}{3})^{\gamma_{ij}} \geq \frac{n-1}{n}(\frac{4}{3})^{\gamma}$, and the equality can be achieved if and only if $\gamma_{ij} = \gamma$, for all $i \neq j$. Define a function $g(\gamma_1, \ldots, \gamma_n) = -\frac{2}{n}\sum_{i=1}^{n}(\frac{10}{9})^{\gamma_i} + \frac{1}{n^2}\sum_{i=1}^{n}(\frac{4}{3})^{\gamma_i}$. It suffices to prove that $[-2(\frac{10}{9})^{\mu} + \frac{1}{n}(\frac{4}{3})^{\mu}]$ is the lower bound of $g(\gamma_1, \ldots, \gamma_n)$, and the lower bound can be achieved if and only if $\gamma_i = \mu$ for all $i = 1, 2, \ldots, n$.

For any n integers $\gamma_1, \ldots, \gamma_n$ with $\sum_{i=1}^{n}\gamma_i = \frac{2mn}{3}$ and $\gamma_1 \geq \mu + 1 \geq \mu - 1 \geq \gamma_2 \geq 0$, we have $\gamma_1 - 1 \geq \mu$ and $0 \leq \gamma_2 \leq \gamma_1 - 1$. But

$$\begin{aligned} &g(\gamma_1, \gamma_2, \gamma_3, \ldots, \gamma_n) - g(\gamma_1 - 1, \gamma_2 + 1, \gamma_3, \ldots, \gamma_n) \\ &\quad = \frac{1}{n^2}\left[\frac{1}{3}\left(\frac{4}{3}\right)^{\gamma_1 - 1} - \frac{2n}{9}\left(\frac{10}{9}\right)^{\gamma_1 - 1} - \frac{1}{3}\left(\frac{4}{3}\right)^{\gamma_2} + \frac{2n}{9}\left(\frac{10}{9}\right)^{\gamma_2}\right] \\ &\quad = \frac{1}{n^2}\left(f(\gamma_1 - 1) - f(\gamma_2)\right). \end{aligned}$$

Under the condition $f(\mu) \geq f(0)$, we will always have $f(\gamma_1 - 1) - f(\gamma_2) \geq 0$, so the assertion is then obvious. □

Theorem 1 can only deal with the situations when μ and γ are both integers. For the more general case, the following theorem holds.

Theorem 2. *For a U-type design $U(n, 3^m)$, let $\mu = \lfloor \frac{2}{3}m \rfloor$, $\gamma = \lfloor \frac{2m(n-3)}{9(n-1)} \rfloor$, and denote $n_\mu = (\mu + 1)n - \frac{2mn}{3}$ and $n_\gamma = (\gamma + 1)\frac{n(n-1)}{2} - \frac{mn(n-3)}{9}$. If $f(\mu) \geq f(0)$, then*

$$\begin{aligned} (CD_2(\mathcal{P}))^2 \geq\ & \left(\frac{13}{12}\right)^m - \frac{2}{n}\left[n_\mu\left(\frac{10}{9}\right)^{\mu} + (n - n_\mu)\left(\frac{10}{9}\right)^{\mu+1}\right] \\ & + \frac{1}{n^2}\left[n_\mu\left(\frac{4}{3}\right)^{\mu} + (n - n_\mu)\left(\frac{4}{3}\right)^{\mu+1}\right] \\ & + \frac{2}{n^2}\left[n_\gamma\left(\frac{4}{3}\right)^{\gamma} + \left(\frac{n(n-1)}{2} - n_\gamma\right)\left(\frac{4}{3}\right)^{\gamma+1}\right]. \end{aligned} \tag{2.3}$$

The lower bound can be obtained if and only if n_μ γ_i*'s take the value* μ, $n - n_\mu$ γ_i*'s take the values* $\mu+1$*, and if, for all* $i \neq j$, n_γ γ_{ij}*'s take the value* γ *and* $\frac{n(n-1)}{2} - n_\gamma$ γ_{ij}*'s take the value* $\gamma + 1$.

The technique to prove Theorem 2 is essentially the same as that for Theorem 1. However, note that when $\gamma_1 \geq \mu + 2 \geq \mu \geq \gamma_2 \geq 0$, we will also have $\gamma_1 - 1 \geq \mu$ and $0 \leq \gamma_2 \leq \gamma_1 - 1$. Similarly, it is easy to prove that when n_γ γ_{ij}'s take the value γ and $\frac{n(n-1)}{2} - n_\gamma$ γ_{ij}'s take the value $\gamma + 1$ for all $i \neq j$, then $\sum_{i \neq j} (\frac{4}{3})^{\gamma_{ij}}$ achieves its minimum.

According to the property of $f(x)$, the condition $f(\mu) \geq f(0)$ is trivial when the zero point of $\frac{df(x)}{dx}$, x_1, is less than or equal to zero. When $x_1 > 0$, the condition $f(\mu) \geq f(0)$ is actually a constraint to m and n. In Table 1, we list the smallest m which satisfies the constraint for different given (small) values of n. Table 2 provides the numerical values of the lower bounds for some three-level designs where this requirement holds.

Remark. When the condition $f(\mu) \geq f(0)$ is not satisfied, then Theorem 2 is not valid. For example, for $n = 12$ and $m = 10$, when all γ_{ij}'s take 1 or 2, and all γ_i's take 6 or 7, the CD_2 value is 0.312198. However, there does exist another $U(12; 3^{10})$ with CD_2 value 0.311965.

Now, consider the four-level case. Similar to the three-level case, the expression $\alpha_i^l = 1 + \frac{1}{2}|x_{il} - \frac{1}{2}| - \frac{1}{2}|x_{il} - \frac{1}{2}|^2$ can only take two possible values, i.e., $\frac{143}{128}$ or $\frac{135}{128}$, while the expression $\beta_{ij}^l = 1 + \frac{1}{2}|x_{il} - \frac{1}{2}| + \frac{1}{2}|x_{jl} - \frac{1}{2}| - \frac{1}{2}|x_{il} - x_{jl}|$ can take three possible values, i.e., 1, $\frac{11}{8}$ or $\frac{9}{8}$. Consequently, the analytical expression (1.1) can be simplified to

$$
\begin{aligned}
(CD_2(\mathcal{P}))^2 = {} & \left(\frac{13}{12}\right)^m - \frac{2}{n}\sum_{i=1}^{n}\left(\frac{143}{128}\right)^{\gamma_i}\left(\frac{135}{128}\right)^{m-\gamma_i} \\
& + \frac{1}{n^2}\sum_{i=1}^{n}\left(\frac{11}{8}\right)^{\gamma_i}\left(\frac{9}{8}\right)^{m-\gamma_i} + \frac{1}{n^2}\sum_{i\neq j}\left(\frac{11}{8}\right)^{\gamma_{ij}^{(1)}}\left(\frac{9}{8}\right)^{\gamma_{ij}^{(2)}},
\end{aligned} \tag{2.4}
$$

where $\gamma_i = \sharp\{x_{il} = \frac{1}{8} \text{ or } \frac{7}{8}, l = 1, \ldots, m\}$, $\gamma_{ij}^{(1)} = \sharp\{(x_{il}, x_{jl}) \in \{(\frac{1}{8}, \frac{1}{8}), (\frac{7}{8}, \frac{7}{8})\}$, $l = 1, \ldots, m\}$, and $\gamma_{ij}^{(2)} = \sharp\{(x_{il}, x_{jl}) \in \{(\frac{3}{8}, \frac{3}{8}), (\frac{5}{8}, \frac{5}{8}), (\frac{1}{8}, \frac{3}{8}), (\frac{5}{8}, \frac{7}{8})\}, l = 1, \ldots, m\}$.

Similar to the three-level case, we can make use of the arithmetic and geometric mean inequality to prove that the lower bound of

$$
\frac{1}{n^2}\sum_{i\neq j}\sigma_{ij} = \frac{1}{n^2}\sum_{i\neq j}\left(\frac{11}{8}\right)^{\gamma_{ij}^{(1)}}\left(\frac{9}{8}\right)^{\gamma_{ij}^{(2)}}
$$

is $\frac{n-1}{n}e^{\tilde{\delta}}$, where

$$
\tilde{\delta} = \frac{m(n-4)}{8(n-1)}\ln\left(\frac{11}{8}\right) + \left(\frac{m(n-4)}{8(n-1)} + \frac{mn}{4(n-1)}\right)\ln\left(\frac{9}{8}\right),
$$

TABLE 1. Smallest m for given n (three-level designs)

n	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57
m	6	11	14	17	18	20	21	22	23	24	25	26	27	27	28	28	29	29

TABLE 2. Lower bounds for selected three-level and four-level designs

(a) $U(n, 3^m)$

n \ m	2	3	4	5	6	7	8	9	10	11	12	13
3	0.029578	0.061535	0.116275	0.191630	0.288337	0.431354	0.612071	0.831422	1.140121	1.516970	1.963161	2.576663
6	–	–	–	–	0.150477	0.213476	0.300838	0.413498	0.563481	0.746932	0.965041	1.257973
9	–	–	–	–	–	–	–	–	–	0.514944	0.657025	0.865048

n \ m	14	15	16	17	18	19	20	21	22	23	24
3	3.312508	4.172209	5.339566	6.725619	8.332293	10.498205	13.054053	16.002281	19.959493	24.610902	29.959613
6	1.611589	2.027403	2.602909	3.287498	4.083090	5.145613	6.403199	7.858293	9.802713	12.105863	14.770851
9	1.113203	1.403003	1.781226	2.231992	2.757224	3.481214	4.335311	5.321957	6.624710	8.159739	9.930152
12	0.872241	1.090803	1.370384	1.720700	2.143674	2.687424	3.328801	4.070250	5.041195	6.230572	7.597594
15	–	–	–	1.431483	1.775544	2.211150	2.724895	3.319226	4.137908	5.096484	6.198060
18	–	–	–	–	1.530124	1.893633	2.322292	2.847807	3.535717	4.340425	5.265036
21	–	–	–	–	–	–	2.034718	2.511079	3.105581	3.800382	4.598591
24	–	–	–	–	–	–	–	2.258533	2.782978	3.395351	4.098757

(b) $U(n, 4^m)$

n \ m	2	3	4	5	6	7	8	9	10	11	12	13
4	0.015028	0.039476	0.076399	0.137246	0.219310	0.342198	0.500241	0.726006	1.008982	1.402187	1.887314	2.549491
8	–	0.016824	0.033722	0.060583	0.098319	0.153966	0.227347	0.331473	0.464204	0.648181	0.877905	1.191364
12	–	–	–	–	0.063308	0.098902	0.147023	0.214248	0.301322	0.420864	0.571762	0.776417
16	–	–	–	–	–	–	–	0.158017	0.223036	0.311303	0.423922	0.575534
20	–	–	–	–	–	–	–	–	–	0.246828	0.336837	0.457050
24	–	–	–	–	–	–	–	–	–	–	–	0.378899

n \ m	14	15	16	17	18	19	20	21	22	23	24
4	3.357991	4.448174	5.769673	7.536213	9.666549	12.496404	15.896181	20.391240	25.776497	32.871740	41.354301
8	1.577482	2.098529	2.734354	3.585419	4.617007	5.989415	7.644773	9.836817	12.471137	15.947054	20.112780
12	1.030449	1.371919	1.790923	2.350361	3.031238	3.935607	5.029766	6.477193	8.220680	10.519735	13.279906
16	0.765134	1.018718	1.331545	1.747779	2.256337	2.930206	3.747833	4.827634	6.131085	7.847874	9.912312
20	0.608502	0.809960	1.059795	1.390946	1.797092	2.333804	2.986836	3.847597	4.888796	6.258262	7.907600
24	0.505130	0.672090	0.880227	1.155005	1.493287	1.939040	2.482900	3.198280	4.065392	5.204171	6.577768
28	0.431803	0.574242	0.752741	0.987425	1.277434	1.658451	2.124605	2.736460	3.479593	4.454023	5.631153
32	–	0.501201	0.657550	0.862256	1.116171	1.448765	1.856790	2.391176	3.041529	3.892930	4.922982

and that this lower bound can be achieved when all σ_{ij}'s take the value $e^{\tilde{\delta}}$, for $i \neq j$.

Now define

$$\begin{aligned} g^*(\gamma_1, \gamma_2, \ldots, \gamma_n) &= -\frac{2}{n}\sum_{i=1}^{n}\left(\frac{143}{128}\right)^{\gamma_i}\left(\frac{135}{128}\right)^{m-\gamma_i} + \frac{1}{n^2}\sum_{i=1}^{n}\left(\frac{11}{8}\right)^{\gamma_i}\left(\frac{9}{8}\right)^{m-\gamma_i} \\ &= -\frac{2}{n}\left(\frac{135}{128}\right)^{m}\sum_{i=1}^{n}\left(\frac{143}{135}\right)^{\gamma_i} + \frac{1}{n^2}\left(\frac{9}{8}\right)^{m}\sum_{i=1}^{n}\left(\frac{11}{9}\right)^{\gamma_i}. \end{aligned}$$

Utilizing the same technique as in the case of $q = 3$, we define a function

$$f^*(x) = \frac{2}{9n^2}\left(\frac{9}{8}\right)^{m}\left(\frac{11}{9}\right)^{x} - \frac{16}{135n}\left(\frac{135}{128}\right)^{m}\left(\frac{143}{135}\right)^{x}.$$

We find that

$$x_0^* = \frac{\log\left(\frac{8n}{15}\right) + m\log\left(\frac{15}{16}\right)}{\log\left(\frac{15}{13}\right)}$$

is the only root of $f^*(x)$, and

$$x_1^* = \frac{\log\left(\frac{8n}{15}\right) + m\log\left(\frac{15}{16}\right) + \log\log\left(\frac{143}{135}\right) - \log\log\left(\frac{11}{9}\right)}{\log\left(\frac{15}{13}\right)}$$

is the only root of $\frac{df^*(x)}{dx}$. It is easy to see that $\lim_{x\to-\infty} f^*(x)=0$ and $\lim_{x\to\infty} f^*(x) = \infty$. The function $f^*(x)$ is strictly decreasing on $(-\infty, x_1]$, and strictly increasing on $[x_1, \infty)$.

Denoting $\mu^* = \lfloor \frac{m}{2} \rfloor$, and using a similar proof as for Theorem 1, we obtain that when $f^*(\mu^*) \geq f^*(0)$, the minimum of $g^*(\gamma_1, \gamma_2, \ldots, \gamma_n)$ can be achieved when all γ_i's equal μ^* or $\mu^* + 1$. Gathering these results, we find

Theorem 3. *For a U-type design $U(n, 4^m)$, let $\mu = \lfloor \frac{m}{2} \rfloor$, and $n_\mu = (\mu+1)n - \frac{mn}{2}$. If $f^*(\mu) \geq f^*(0)$, then*

$$\begin{aligned} &(CD_2(\mathcal{P}))^2 \\ &\quad \geq \left(\frac{13}{12}\right)^m - \frac{2}{n}\left[n_\mu\left(\frac{135}{128}\right)^m\left(\frac{143}{135}\right)^\mu + (n-n_\mu)\left(\frac{135}{128}\right)^m\left(\frac{143}{135}\right)^{\mu+1}\right] \\ &\quad\quad + \frac{1}{n^2}\left[n_\mu\left(\frac{9}{8}\right)^m\left(\frac{11}{9}\right)^\mu + (n-n_\mu)\left(\frac{9}{8}\right)^m\left(\frac{11}{9}\right)^{\mu+1}\right] + \frac{n-1}{n}e^{\tilde{\delta}}. \end{aligned} \tag{2.5}$$

The lower bound can be obtained if and only if n_μ γ_i's take the value μ, $n - n_\mu$ γ_i's take the values $\mu + 1$, and all γ_{ij}'s take the value $e^{\tilde{\delta}}$, for $i \neq j$.

The condition $f^*(\mu^*) \geq f^*(0)$ imposes a constraint on m and n. Table 3 lists the smallest m, which satisfy the constraint for different (small) values of n. The numerical values for some four-level designs are provided in Table 2.

TABLE 3. Smallest m for given n (four-level designs)

n	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72
m	1	3	6	9	11	13	14	15	17	17	18	19	20	20	21	22	22	23

Theorems 1–3 provide lower bounds for uniform designs with three and four levels, respectively, which can serve as benchmarks for judging the quality of a U-type design with regard to the centered L_2-discrepancy CD_2. Besides serving as a benchmark, these theorems also provide a direct characterization of U-type designs which are uniform designs. In fact, the above discussion has shown that under certain conditions, when the distributions of certain variables become as even as possible, the CD_2 value of the design can be expected to achieve the minimum. We will use this result in the next section when introducing a new optimization algorithm named balance-pursuit heuristic. Instead of using a purely stochastic search approach like threshold accepting, this new algorithm uses a more guided search. Consequently, it is expected to provide fast approximations to uniform designs. We also compare its performance with different versions of the threshold accepting algorithm to find out when we can expect a better performance.

3. Balance-pursuit heuristic algorithm

The threshold accepting heuristic has been successfully applied to generate low-discrepancy and uniform designs [4, 5, 6]. In particular, many known uniform and almost uniform designs, especially those listed on the web site http://www.math.hkbu.edu.hk/UniformDesign/, have been obtained by using several implementations of threshold accepting. The introduction of new lower bounds in Section 2 for the CD_2-discrepancy for the three- and four-level cases allows us to extend the analysis to these cases and to evaluate new designs.

However, the results presented in the previous section also provide information about the characteristics of low-discrepancy and uniform designs. Thus, it might be helpful to use this information for a different optimization approach. While the previous implementations of the threshold accepting algorithm use a random approach for generating new candidate solutions, efficient use of the additional information might result in a more directed search. This idea is implemented in a new algorithm named balance-pursuit heuristic which is also a sequential procedure. However, new candidate solutions are generated taking into account the conditions for uniform designs provided in Theorems 1–3. This approach will be discussed in more detail in Subsection 3.2.

In this section, we introduce and discuss several modifications of threshold accepting and the balance-pursuit heuristic. The following section provides a comparison of the performance of these different implementations depending on the size of the designs under study and the number of iterations, i.e., the computational resources used.

3.1. **Lower bounds.** All algorithms considered are local search heuristics, but differ in the method for generating new candidate solutions and the acceptance criterion. The pseudo-code of a prototype local search heuristic is provided in Algorithm 1. The algorithm starts with a randomly generated U-type design D^c. Then a finite large number, say τ, is used to determine the number of iterations performed during a single optimization run. The algorithm stops as soon as the lower bound is reached (step 4). Given that the lower bound cannot be obtained for all parameters, and since a good approximation might be sufficient for many applications, step 4 of the algorithm could be replaced by a relaxed condition, e.g., $D^{\min} > (\text{lower bound}) \times 1.01$, to allow for deviations of up to one percent. In both cases, the knowledge about the lower bound helps to reduce computational time if a

satisfying solution is found early. As long as this is not the case, in each iteration the algorithm tries to replace the current solution D^{c} with a new solution D^{new}. The new design is generated in a given neighborhood of the current solution D^{c}. Obviously, D^{new} should also be a U-type design. Therefore, generating new candidate solutions is equivalent to imposing a neighborhood structure on the search space $\mathcal{U}(n; q^m)$. In Subsection 3.2 we will discuss several neighborhood structures and show how the findings from Theorems 1–3 translate to such structures. Closely related to this issue is the idea of local updating of the objective function when comparing the discrepancy of two candidate designs. This issue will also be discussed in the following subsection, while Subsection 3.3 is devoted to the different acceptance criteria and implementation details related to it.

Algorithm 1 Pseudo code for prototype local search heuristic.

```
Initialize τ (number of iterations)
Generate starting design D^c ∈ U(n; q^m) and let D^min := D^c
for i = 1 to τ do
   while D^min > lower bound do
      Generate D^new ∈ N(D^c)    (neighbor to current solution)
      Compute ∇ = CD_2(D^new) − CD_2(D^c) and generate u (uniform random variable)
      if (∇ < 0) or acceptance criterion(∇, u) met then D^c = D^new
      if D^c < D^min then D^min := D^c
   end while
end for
```

3.2. **Neighborhoods.** The definition of neighborhoods $\mathcal{N}$ has to take into account several conditions. First, as already pointed out above, $\mathcal{N}(D^{\text{c}}) \subset \mathcal{U}(n; q^m)$, i.e., each D^{new} should also be a U-type design. Second, in order to impose a real "local" structure, the designs D^{c} and those in $\mathcal{N}(D^{\text{c}})$ should not differ too much. Third, the computational complexity of the algorithm depends to a large extent on calculating ∇, i.e., the difference in the objective function when moving from D^{c} to D^{new}. Thus, if ∇ can be obtained without calculating $CD_2(D^{\text{new}})$ from scratch, a significant speed-up might result.

All three requirements can easily be fulfilled by selecting one or more columns of D^{c} and exchanging two elements within each selected column. Then, the first condition is satisfied by construction. The second condition is also obviously fulfilled if the number of exchanges is limited to a low number relative to the size of the design.

For the third condition, assume that D^{new} is obtained from D^{c} by exchanging the elements in rows i and j of column k. We know that the centered L_2-discrepancy can be expressed as a function of the sum of the σ_{ij}'s and the σ_i's as defined in equation (1.2). Now, for a single exchange of two elements in the selected column, there are altogether $2(n-2)$ σ_{ij}'s and two σ_i's which have to be updated. Suppose in column k the elements in rows i and j are exchanged. Then for any row l other than i or j, denote $\tilde{\sigma}_{il}$ and $\tilde{\sigma}_{jl}$ as the new distances between row pair (i, l) and row pair (j, l), and denote $\tilde{\sigma}_i = -\tilde{\sigma}_i^{(1)} + \tilde{\sigma}_i^{(2)}$ and $\tilde{\sigma}_j = -\tilde{\sigma}_j^{(1)} + \tilde{\sigma}_j^{(2)}$ as the new distances of the single rows i and j from the centered point $(1/2, 1/2, \ldots, 1/2)$. Then

$$\tilde{\sigma}_i^{(1)} = \sigma_i^{(1)} \cdot \alpha_j^k / \alpha_i^k, \qquad \tilde{\sigma}_j^{(1)} = \sigma_j^{(1)} \cdot \alpha_i^k / \alpha_j^k,$$
$$\tilde{\sigma}_i^{(2)} = \sigma_i^{(2)} \cdot \beta_{jj}^k / \beta_{il}^k, \qquad \tilde{\sigma}_j^{(2)} = \sigma_j^{(2)} \cdot \beta_{ii}^k / \beta_{jj}^k,$$
$$\tilde{\sigma}_{il} = \sigma_{il} \cdot \beta_{jl}^k / \beta_{il}^k, \qquad \tilde{\sigma}_{jl} = \sigma_{jl} \cdot \beta_{il}^k / \beta_{jl}^k,$$

with β_{il}^k, β_{jl}^k and α_i^k, α_j^k as defined in Section 2. The values for σ_k and σ_{kl} for $k, l \neq i, j$ are the same for D^{new} and D^{c}. Using expression (1.3), $\nabla = CD_2(D^{\text{new}}) - CD_2(D^{\text{c}})$ can be rewritten as

$$\begin{aligned}\nabla &= \frac{1}{n^2} \cdot (\tilde{\sigma}_i - \sigma_i + \tilde{\sigma}_j - \sigma_j) + \frac{1}{n^2} \cdot \sum_{t \neq i,j} 2 \cdot (\tilde{\sigma}_{ti} - \sigma_{ti} + \tilde{\sigma}_{tj} - \sigma_{tj}) \\ &= \frac{1}{n^2} \cdot \left(-\sigma_i^{(1)} \cdot \left(\frac{\alpha_j^k}{\alpha_i^k} - 1 \right) - \sigma_j^{(1)} \cdot \left(\frac{\alpha_i^k}{\alpha_j^k} - 1 \right) \right) + \cdots \\ &\quad + \frac{1}{n^2} \cdot \left(\sigma_i^{(2)} \cdot \left(\frac{\beta_{jj}^k}{\beta_{ii}^k} - 1 \right) + \sigma_j^{(2)} \cdot \left(\frac{\beta_{ii}^k}{\beta_{jj}^k} - 1 \right) \right) + \cdots \\ &\quad + \frac{1}{n^2} \cdot \sum_{t \neq i,j} 2 \cdot \left(\sigma_{ti} \cdot \left(\frac{\beta_{tj}^k}{\beta_{ti}^k} - 1 \right) \right) .\end{aligned}$$

If more than one column are exchanged, i.e., $k = k_1, \ldots, k_n$, then replace $\frac{\alpha_j^k}{\alpha_i^k}$ by $\prod_\kappa \frac{\alpha_j^{k_\kappa}}{\alpha_i^{k_\kappa}}$ etc. The updating formula for ∇ avoids calculating $CD_2(D^{\text{new}})$ from scratch prior to the acceptance decision. Furthermore, if D^{new} is accepted as a new current solution, the values of the α's, β's, σ_{ij}'s and σ_i's have to be updated only for some row indices.

The neighborhood definitions used in this paper for the different version of the threshold accepting heuristic and the balance-pursuit heuristic differ in the selection of rows i and j for the exchange in a given column k. We consider seven different cases:

random ("rnd"): randomly select i, j;
preselection 1 ("ps-1"): $w_i = \sum_{l \neq i} \sigma_{il}$; select i s.t. $\arg\max_i(w_i)$ and j s.t. $\arg\min_j(w_j)$;
preselection 2 ("ps-2"): $w_i = \sigma_i$; select i s.t. $\arg\max_i(w_i)$ and j s.t. $\arg\min_j(w_j)$;
roulette wheel ("rw"): $w_i = \sum_{l \neq i} \sigma_{il}$; select i with probabilities based on w_i; select j with probabilities based on w_i^{-1};
strict roulette wheel ("srw"): $w_i = \sum_{l \neq i} \sigma_{il}$; select i with probabilities based on $(w_i - \min(w))$; select j with probabilities based $(w_i - \max(w))^{-1}$;
preselection 3 ("ps-3"): find pair(s) where σ_{il} is maximal and where $\sigma_{j\iota}$ is minimal; select i and j;
mixed preselection ("ps-mx"): in each iteration, one of the preselection methods ps-1, ps-2, and ps-3 is randomly chosen.

In previous applications of threshold accepting to uniform design problems, in general, the rows i and j have been chosen randomly ("rnd"). For the current application, we also consider the use of additional information about search directions which might be more promising. Based on the results provided in Theorems 2 and 3, for the cases $q = 3$ and 4, respectively, we define three preselection methods. The two guided preselection methods ps-1 and ps-3 aim at a uniform distribution of the γ_{ij}'s, while the preselection method ps-2 targets the distribution of the γ_i's. Taking into account that $\sigma_{ij} = \left(\frac{4}{3}\right)^{\gamma_{ij}}$ for the three-level case, and $\sigma_{ij} = \left(\frac{11}{8}\right)^{\gamma_{ij}^{(1)}} \left(\frac{9}{8}\right)^{\gamma_{ij}^{(2)}}$ for the four-level case, the first goal is equivalent to obtaining equal values for all

σ_{ij}. Realizing that $\sigma_i^{(1)} = 2n\left(\frac{10}{9}\right)^{\gamma_i}$ and $\sigma_i^{(2)} = \left(\frac{4}{3}\right)^{\gamma_i}$ for the three-level case, while for the four-level case, $\sigma_i^{(1)} = 2n\left(\frac{143}{128}\right)^{\gamma_i}\left(\frac{135}{128}\right)^{m-\gamma_i}$ and $\sigma_i^{(2)} = \left(\frac{11}{8}\right)^{\gamma_i}\left(\frac{9}{8}\right)^{m-\gamma_i}$, $\sigma_i = -\sigma_i^{(1)} + \sigma_i^{(2)}$ represents the distance of a single row i from the central point $(1/2, 1/2, \ldots, 1/2)$.

The preselection method "ps-1" is called "*single row with maximal and minimal sum of distances*". Based on the distances for all pairs of rows of the current design D^{c}, we identify a single row with maximal and another single row with minimal sum of distances, say row i and row j. This means $\sum_{l\neq i}\sigma_{il}$ is maximal and $\sum_{l\neq j}\sigma_{jl}$ is minimal among $\sum_{l\neq\iota}\sigma_{l\iota}, \iota = 1, \ldots, m$. The preselection method "ps-2" is called "*single row with average distribution*". Then, the rows with maximal and minimal σ_i's, respectively, are selected. Since these rows are not unique, the neighborhood becomes larger by admitting any row with maximal or minimal σ_i's, respectively.

Since the preselection methods "ps-1" and "ps-2" define rather small neighborhoods, we also consider two mixed cases, where the selection of rows i and j, though being random in principle, is biased in favor of row pairs satisfying the condition of the method preselection 1. These "roulette wheel" methods differ only by including ("rw") or excluding ("srw") the "worst" pair of rows.

The preselection method ps-3 is called "*maximal and minimal distances of row pairs*". Denote by (i, l) and (j, ι) the two row pairs with maximal and minimal distances (σ_{ij}) for the current design D^{c}. Then, the candidate rows are i and j. In passing, note that the argument is symmetric in i and l as well as in j and ι.

For all neighborhood selection methods, the selection process of i and j plus the random choice of k are repeated until the obvious condition for a change, $x_{ik} \neq x_{jk}$, is met. A straightforward extension for all neighborhood definitions consists of allowing for exchanging entries in more than one column. Again, this leads to larger neighborhoods, which should be considered in particular for larger designs.

Of course, for the construction of neighborhoods, we are not restricted to using only a single of these methods. Instead, several methods can be combined, e.g., by randomly selecting out of a set of methods in each iteration as proposed for the mixed preselection method ("ps-mx"). This corresponds to forming larger neighborhoods. We will come back to this argument in Section 4.

3.3. **Acceptance.** In each iteration, the algorithm will generate a new candidate solution D^{new} in the neighborhood of the current solution D^{c} (step 5). If this new candidate solution results in an improved value of the centered L_2-discrepancy, i.e., if $\nabla < 0$, the new candidate solution becomes the current solution according to the first argument in step 7 of the prototype Algorithm 1. However, a simple greedy method accepting only improvements of the objective function will typically get stuck in a local minimum of poor quality unless the neighborhoods are defined very large. Thus, step 7 also provides a second option for accepting D^{new} even if it corresponds to an impairment of the objective function. Again, we consider two different approaches:

TA ("TA"): In the threshold accepting implementation, a threshold sequence is initialized. The value of this sequence, which is reduced on a regular basis, determines to what extent a worsening of the objective function is accepted in the acceptance step (step 7). For the present application,

we use a geometric threshold sequence, which is determined by the initial and the terminal threshold value. Furthermore, with a low probability, the elitist solution, i.e., the design $D^{\min}$, is reinforced.

Greedy with mutation ("Greedy"): A less complicated acceptance criterion is used for the second method, which is a nearly strict downhill search, when almost exclusively improvements are accepted in step 7, while impairments are accepted with a rather low probability in order to keep a chance to escape a poor quality local minimum.

While the second method is already completely described by adding a value for the number of iterations τ, which takes on values from 1 000 to 500 000 in our computational experiments, we have to provide some additional information on the threshold sequence for Threshold Accepting. For finding a reasonable threshold sequence, several approaches have been discussed in the literature [14]. One has to take into account that the typical size of local changes ∇ in the fitness function depends strongly on the neighborhood definition. Thus, a standard approach consists of generating a large set of random designs plus a neighboring design according to a given neighborhood definition. Then, calculating the ∇'s for these pairs and evaluating their distribution can provide a first idea. In a next step, different initial values and terminal values (which were considered reasonable based on these ∇'s) were tested, and eventually the values $T_0 = 0.01$ and $T_\tau = 0.00001$ were chosen for the initial and terminal threshold, respectively. The threshold is lowered geometrically after every 200 iterations by a factor such that the terminal value is met for the last 200 iterations.[2]

Finally, the reinforcement of the elitist, the elitist principle, is implemented for the threshold accepting algorithm as follows. If a change has been rejected in step 7, then with a low probability π_ε the current solution is replaced with the best solution found so far. Though the algorithm appears rather insensitive to its value, preliminary tests suggested that $\pi_\varepsilon = 0.000005$ is a reasonable general value. Ideally, π_ε ought to be adjusted for different numbers of iterations and problem size—yet this idea was abandoned for the sake of simplicity. Also, note that π_ε is very low and that in many runs the current solution is never replaced with the elitist; this applies in particular for the runs with a low number of iterations.

4. Comparison of methods and results

In order to analyze the performance of the algorithms using different neighborhood definitions and acceptance criteria, we first concentrate on two examples, i.e., the designs $U(39; 3^{10})$ and $U(99; 3^{10})$ for which a total of approximately 3 000 and 4 000 runs were performed, respectively. The implementation for this comparative study was done in Matlab R13 on Pentium IV 2.7GHz machines. A more comprehensive set of designs has been analyzed using those methods which appeared most attractive in this comparison. These results for the three- and four-level cases are reported in Subsection 4.2. Detailed up-to-date information on designs can be found on the web site http://www.math.hkbu.edu.hk/UniformDesign/.

[2]Alternatively, we also considered a linear threshold sequence with terminal threshold equal to zero, but did not observe any apparent performance difference to the geometric scheme.

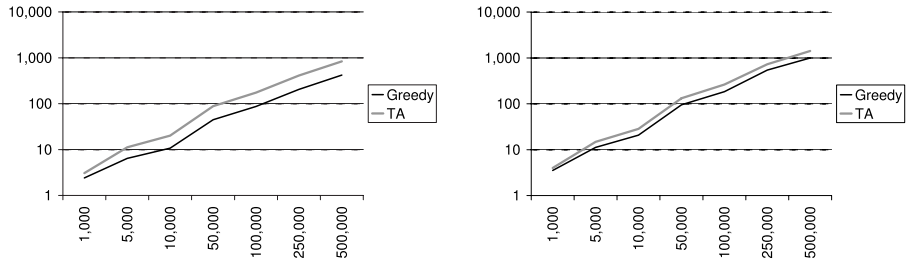

FIGURE 2. Average CPU time in seconds for different number of iterations for $U(99; 3^{10})$

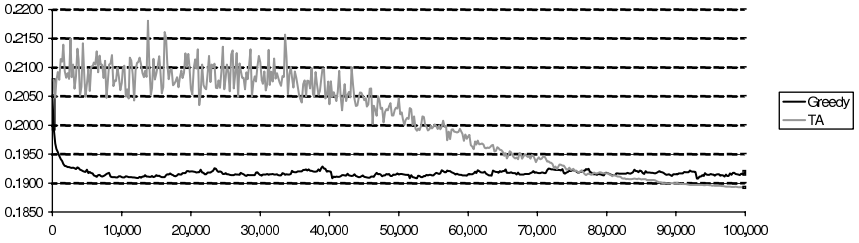

FIGURE 3. Quality of current designs in the course of iterations from two specimen runs for the $U(99; 3^{10})$ design

4.1. **Performance analysis.**

Acceptance criteria. As described earlier, we consider two alternative acceptance criteria in the search process: While "Greedy" has a strict preference for improving steps and allows for mutations (i.e., possible impairments) only with a very low probability, TA is more tolerant towards impairments as long as they do not exceed a certain level. This implies that TA accepts more changes during the search process which demand more CPU time as updates are time costly, as Figure 2 illustrates. It plots the mean computational time (in seconds) across neighborhood definitions for both acceptance criteria for different numbers of iterations τ. Three observations are worth noting. First, computational time grows at a *linear rate* with the number of iterations. Second, the computational time is hardly affected by the size of the design under consideration. However, in order to obtain results of similar quality, it might be necessary to increase the number of iterations with the size of the design. As long as this increase is linear or polynomial in the size of the design, the computational complexity of the algorithm remains linear or polynomial, respectively. Third, the additional computational load of TA relative to the Greedy strategy appears to be an almost constant factor (note that the computational time is plotted on a logarithmic scale).

Another advantage of the Greedy strategy is that it has a good chance of identifying an optimum within the current neighborhood as it follows a downhill search strategy. Hence, when the number of iterations, i.e., the CPU time, is limited, then Greedy can be expected to find better results than TA. If, however, there is sufficient CPU time, TA turns out to find better results eventually, as it is better suited to escape local optima. TA therefore benefits more from an increase in CPU time than the Greedy acceptance criterion does. Figure 3 illustrates this by depicting

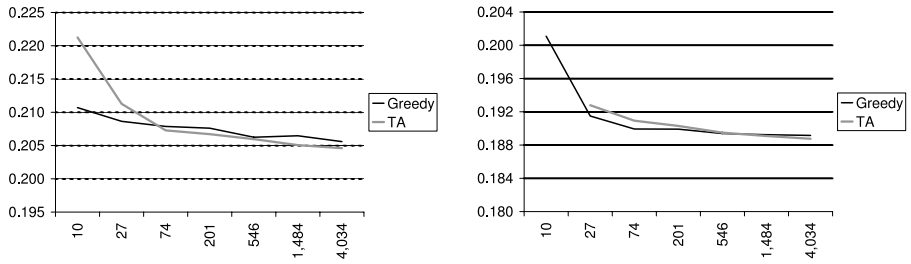

FIGURE 4. Best found solutions for different CPU times (in seconds; logarithmic scale)

the square of the $CD(\mathcal{P})$'s of the respective current designs in the course of 100 000 iterations from two specimen runs. In passing, note that for either algorithm, the last iteration's design need not be the optimal solution found in this run, $D^{\min}$, and that $D^{\min}$ is eventually reported. Consequently, there exists a critical amount of computation time, when TA begins to outperform the Greedy approach. The larger the problem size, the higher this critical time is, while for smaller problems, TA finds better results even when the number of iterations is very small (see Figure 4). This figure plots the best found solution for the two acceptance criteria against computation time. The critical time for the smaller problem instance $U(39; 3^{10})$ appears to be about one minute, while this value increases to about 10 minutes for the larger problem instance $U(99; 3^{10})$.

Guided and random search. The suggested neighborhood definitions follow different concepts. While the preselection methods make use of the characterization of uniform designs in Theorems 1–3 in order to derive a search direction, others incorporate stochastic components or are perfectly random. The effect of these neighborhood concepts on the search process is rather diverse. While the random selection ("rnd") requires almost no computational cost, methods that have to evaluate the current situation are of higher computational complexity and therefore take longer. Also, methods where a search step comes with rather low changes in the objective function need more interim update steps which, too, has a negative effect on the computational complexity. Figure 5(a) illustrates this for the $U(99; 3^{10})$ design and 500 000 iterations—along with the aforementioned effect that TA accepts more updates and therefore needs additional time.

The more important aspect, however, is how these preselection methods affect the quality of the results. As can be seen from Figure 5, methods tend to be less reliable the stronger their deterministic component is. Both acceptance criteria have a preference for downhill moves. If this preference is combined with a preselection method that also suggests (supposed) downhill moves, then chances are that the search gets stuck once a local optimum has been reached. In particular, for preselection methods ps-1 and ps-2, the found solutions are on average worse than those of the other methods (Figure 5(c)). Also, the standard deviation of their reported solutions is very high (Figure 5(d)):[3] once they have converged to a local optimum, these two methods have problems escaping them again. Hence, the

[3]Standard derivations are calculated based on 100 independent runs for each specification of the algorithms.

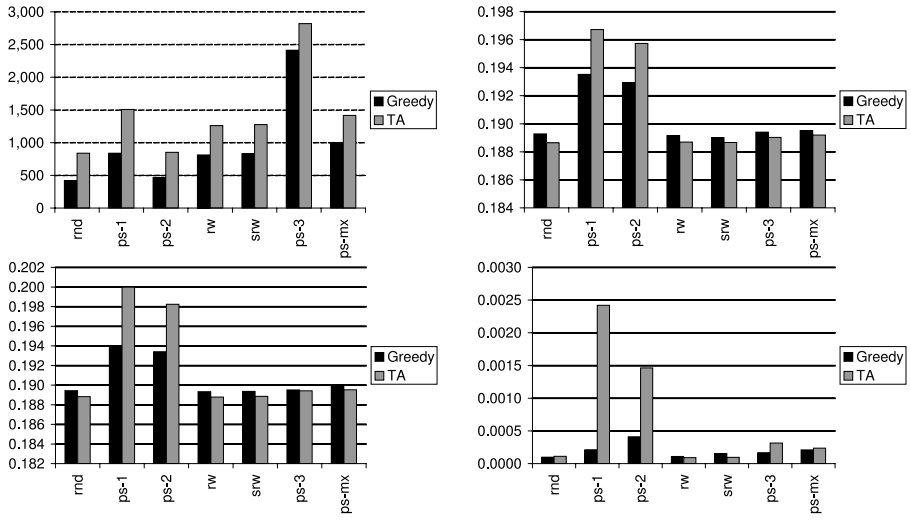

FIGURE 5. Results for the $U(99; 3^{10})$ design and 500 000 iterations

quality of the found solution depends highly on the initial solution of the search process. Apparently, these problems become more evident when TA is used rather than Greedy. One reason might be that the (absolute) values for ∇ are smaller than for other selection methods; with equal threshold sequence, TA will then accept more downhill moves than with methods with large (absolute) values for ∇. This higher acceptance rate therefore increases the computational time and has a negative effect on convergence. Hence, for these particular selection methods the threshold sequence ought to be reconsidered.

When there is a higher stochastic component in the selection method, then the average results become better and more stable. This supports the view that a neighborhood search strategy should consider not only downhill, but also uphill moves. This is all the more true for rough and demanding solution spaces than for this problem. Both acceptance criteria benefit from stochastic components in the selection step. At the same time, it can also be seen for TA that introducing a preference for search steps that supposedly improve the solution (such as the methods ps-1 and ps-2) merely increases the CPU time, but has not necessarily a positive effect on the quality of the found solutions. TA exhibits a rather poor convergence behavior with preselection methods ps-1 and ps-2. Partially, this negative effect could be reduced by introducing refined threshold sequences for this selection method; nonetheless, the achievable improvements appear limited. On the other hand, TA works its best when there is a high degree of randomness in the preselection method: the "rnd" as well as the "rw" and "srw" preselection methods all report good and stable results.

The positive effect of the stochastic component also explains why preselection method ps-3 performs better than methods ps-1 and ps-2: In most cases, there is not a single, but a number of pairs for σ_{il} that share the same maximal (or minimal) value. Hence, this method often makes a stochastic selection. Also, for the mixed strategy, where in each iteration step one of the preselections ps-1, ps-2 or ps-3

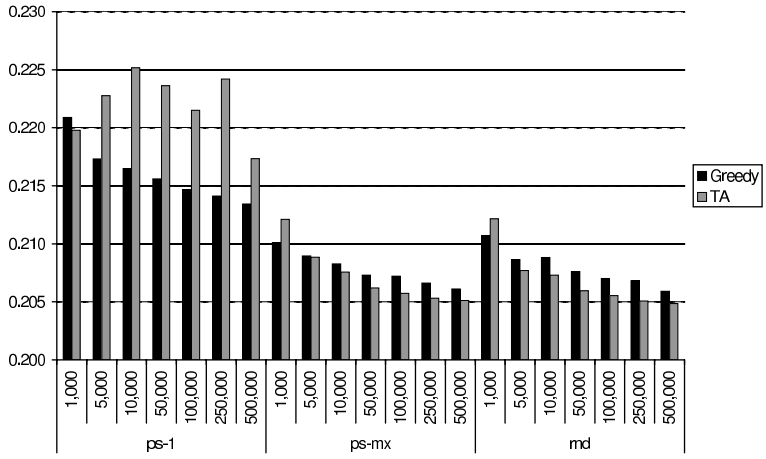

FIGURE 6. Best found solution for $U(39; 3^{10})$

is performed, it is eventually those steps where ps-3 is chosen that contribute the most.

These findings are confirmed by the results from the second test design with 39 rows. As can be seen from Figure 6, a random selection with even a very low number of iterations is able to find better results than a strict deterministic selection rule (in particular preselection 1, yet also the mixed preselection). The results for other designs considered in detail, including four-level designs $U(n, 4^m)$, look similar; a presentation can therefore be left out; results are available upon request. Nevertheless, the difference between the pure random selection and the mixed preselection strategy is not very important. Thus, it cannot be excluded that for different design parameters, the ranking might change. Therefore, for the further calculations on an extended set of three- and four-level designs, we consider both the TA algorithm with the pure random selection and the greedy algorithm with the mixed preselection method.

4.2. **Improved results.**

4.2.1. *Comparison with existing results.* There are many low-discrepancy designs listed in the web site http://www.math.hkbu.edu.hk/UniformDesign/ at this time. Those designs have been obtained by implementing the threshold accepting algorithm. However, by implementing our new specification of the TA algorithm and the greedy balance-pursuit algorithm with the mixed preselection strategy, we find that many of these designs can still be improved. Note that most of these improved designs do not satisfy the constraint on m and n which ensures the condition $f(\mu) \geq f(0)$. So the lower bounds of (2.3) and (2.5) are not valid here, but when the distribution of distances between distinct rows σ_{ij}'s becomes even, the discrepancy of the design is still hoped to be lower.

The results for $q = 3$ summarized in Table 4 have been obtained from independent runs with the Greedy and the TA acceptance criterion. The former approach was implemented in C++ using the Visual Studio environment, the latter using Matlab R13. Some of the algorithms' parameters may differ from those used for comparison in the previous sections, as altered values for the threshold

TABLE 4. Low-discrepancy designs with three levels and comparison with existing results

n	m	Prev. Res.	Results	n	m	Prev. Res.	Results
18	6	0.086931	0.086896	18	7	0.114207	0.113591
18	9	0.194699	0.193463	18	10	0.248099	0.246956
21	6	0.088221	0.088205	21	7	0.114634	0.114446
21	8	0.147239	0.147059	21	9	0.187778	0.187364
21	10	0.237259	0.236923	21	11	0.299072	0.296678
27	7	0.109180	0.108284	27	8	0.139677	0.138657
27	9	0.175623	0.175317	27	10	0.220139	0.220005
27	11	0.273620	0.273468	33	6	0.084048	0.083959
33	7	0.107885	0.107875	33	8	0.136667	0.136571
33	9	0.171391	0.171231	33	10	0.212732	0.212241
33	11	0.261448	0.261221	33	12	0.321370	0.319651
36	7	0.106591	0.106444	36	8	0.134521	0.133659
36	9	0.167871	0.166957	36	10	0.207867	0.206584
36	11	0.255643	0.254961	36	12	0.313376	0.311067
39	6	0.083269	0.083180	39	7	0.106408	0.106296
39	9	0.166801	0.166213	39	10	0.205877	0.204760
39	12	0.305374	0.305317	42	8	0.133118	0.133111
42	9	0.165474	0.165094	42	10	0.203445	0.203321
42	11	0.249071	0.248529	42	12	0.303332	0.302409

Remark: The columns "Prev. Res." indicate the results previously listed on the web site http://www.math.hkbu.edu.hk/UniformDesign/, while the columns "Results" indicate the results obtained by our code.

sequence, mutation probability, etc. proved favorable for different designs. Also, more demanding problems were conceded for a higher number of iterations, typically, 1 500 000 or more. For designs with m exceeding 15, the values of up to three columns rather than of a single column were exchanged per search step.

4.2.2. *Comparison with the lower bound.* According to the lower bound (2.3) in Theorem 2, we implement our algorithms to search the design with parameters row n and column m, which satisfy the constraint $f(\mu) \geq f(0)$. For each row n from 6 to 24, we search designs with up to 24 columns. Table 5 summarizes for which three-level designs the lower bound has been reached by at least one of our algorithms.

TABLE 5. Three-level designs $U(n, 3^m)$ for which the lower bound has been reached ($\surd$) ($-$: no lower bound available; $\circ$ (†; ‡; $\star$): best reported solution deviates 0.1 (0.5; 1; 1.5) percent or less from lower bound)

	number of columns, m																		
n	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
6	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
9	$-$	$-$	$-$	$-$	$-$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
12	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
15	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$\surd$	$\surd$	$\surd$	$\surd$	†	$\surd$	$\surd$	$\surd$
18	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$\surd$	$\surd$	‡	†	$\surd$	$\surd$	$\surd$
21	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$\star$	†	$\circ$	$\circ$	†
24	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$\star$	†	†	‡

5. Conclusion and discussion

In this paper, we investigate the analytical expression of the centered L_2-discrepancy in depth. This analysis provides new lower bounds for three- and four-level U-type designs. Furthermore, these results can be used to characterize low discrepancy designs and to use this characterization for a guided selection of the search direction in optimization heuristics. We compare the performance of a search heuristic based on these findings, called balance-pursuit heuristic, with a multipurpose implementation of the threshold accepting algorithm. Although the guided search does not always outperform the threshold accepting approach, the quality of its results is often similar. In particular, when the computational time is small compared to the design size, the novel approach might be preferred.

We use both the new algorithm and a modified threshold accepting implementation to search for low-discrepancy or even uniform three- and four-level designs. It turns out that quite often we can improve on previous solutions. For the three-level case, we can even obtain many new uniform designs.

Given that the idea for the construction of the lower bounds and the balance-pursuit approach is valid for higher level designs as well, future research will concentrate on generalizing our results for designs with more or even arbitrary levels.

References

[1] R. A. Bates, R. J. Buck, E. Riccomagno, and H. P. Wynn (1996), *Experimental design and observation for large systems*, *Journal of Royal Statistical Society (B)*, **58**, 77–94. MR1379235 (96k:62213)

[2] K. T. Fang, G. N. Ge, M. Q. Liu, and H. Qin (2003), *Construction on minimum generalized aberration designs*, *Metrika*, **57**, 37–50. MR1963710 (2004b:62181)

[3] K. T. Fang, X. Lu, Y. Tang, and J. X. Yin (2004), *Constructions of uniform designs by using resolvable packings and coverings*, *Discrete Mathematics*, **274**, 25–40. MR2025074 (2004k:05057)

[4] K. T. Fang, D. K. J. Lin, P. Winker, and Y. Zhang (2000), *Uniform Design: Theory and Application*, *Technometrics*, **42**, 237–248. MR1801031

[5] K. T. Fang, X. Lu, and P. Winker (2003), *Lower bounds for centered and wrap-around L_2-discrepancies and construction of uniform design by threshold accepting*, *Journal of Complexity*, **19**, 692–711. MR2003240 (2004g:62145)

[6] K. T. Fang, C. X. Ma, and P. Winker (2002), *Centered L_2-discrepancy of random sampling and Latin hypercube design, and construction of uniform designs*, *Mathematics of Computation*, **71**, 275–296. MR1863000 (2002h:65024)

[7] K. T. Fang and Y. Wang (1994), *Number-theoretic methods in statistics*, *Chapman and Hall*, London. MR1284470 (95g:65189)

[8] F. J. Hickernell (1998a), *A generalized discrepancy and quadrature error bound*, *Mathematics of Computation*, **67**, 299–322. MR1433265 (98c:65032)

[9] F. J. Hickernell (1998b), *Lattice rules: how well do they measure up?* in P. Hellekalek and G. Larcher (eds), *Random and Quasi-Random Point Sets*, Springer, 106–166. MR1662841 (2000b:65007)

[10] X. Lu, W. B. Hu, and Y. Zheng (2003), *A systematical procedure in the construction of multi-level supersaturated design*, *Journal of Statistical Planning and Inference*, **115**, 287–310. MR1984066 (2004d:62272)

[11] M. Q. Liu and R. C. Zhang (2000), *Construction of $E(s^2)$ optimal supersaturated designs using cyclic BIBDs*, *Journal of Statistical Planning and Inference*, **91**, 139–150. MR1792369 (2001h:62143)

[12] H. Niederreiter (1992), *Random Number Generation and Quasi-Monte Carlo Methods*, *SIAM CBMS-NSF Regional Conference Series in Applied Mathematics*, Philadelphia. MR1172997 (93h:65008)

[13] Y. Wang and K. T. Fang (1981), *A note on uniform distribution and experimental design*, *KeXue TongBao*, **26**, 485–489. MR0666355 (83k:62109)
[14] P. Winker (2001), *Optimization heuristics in econometrics: Applications of threshold accepting*, *Wiley*, Chichester. MR1883767 (2002k:62011)
[15] P. Winker and K. T. Fang (1998), *Optimal U-type design*, in H. Niederreiter, P. Zinterhof, and P. Hellekalek (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 1996*, Springer, 436–448.

Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong, People's Republic of China
E-mail address: ktfang@math.hkbu.edu.hk

Faculty of Economics, Law and Social Sciences, University of Erfurt, Germany
E-mail address: dietmar.maringer@uni-erfurt.de

Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong, People's Republic of China
Current address: Department of Mathematics, Suzhou University, Suzhou, 215006, People's Republic of China
E-mail address: ytang@math.hkbu.edu.hk

Faculty of Economics, Law and Social Sciences, University of Erfurt, Germany
E-mail address: peter.winker@uni-erfurt.de