Constructing low star discrepancy point sets with genetic algorithms

Carola Doerr, CNRS & LIP6, Paris

joint work with François-Michel De Rainville, Université Laval, Québec
Star Discrepancy – Recap and Notation

- Given: a set of points
  \[ X = \{x^1, ..., x^n\} \]
  with \( x^i \in [0,1)^d, i \in [n] \)

- For all \( y \in [0,1]^d \), local discrepancy
  \[ d^*_\infty (y, X) := V_y - \frac{A(y, X)}{n} \]
  where \( V_y := \prod_{j=1}^d y_j \)
  and \( A(y, X) := |X \cap [0, y)| \)

- The star discrepancy of \( X \) is
  \[ d^*_\infty (X) := \sup_{y \in [0,1]^d} |d^*_\infty (y, X)| \]
What to Expect from This Talk

- **Computer Science/Practitioners’ perspective:**
  focus on single configurations, not on asymptotic behavior
  point sets, not on sequences

- **Main goal:**
  - discrepancy optimization: for given dimension $d$ and given number of points $n$, find $x^1, \ldots, x^n$ in $[0,1)^d$ such that $d^*_\infty(x^1, \ldots, x^n)$ is minimized
    (relevant for applications with fixed budget)
  - inverse discrepancy optimization: for given $d$ and discrepancy bound $\varepsilon$, find smallest possible $n$ and points $x^1, \ldots, x^n$ in $[0,1)^d$ such that $d_\infty(x^1, \ldots, x^n) \leq \varepsilon$
    (relevant for applications with maximal allowed precision error)

- **Typical application:** industrial or academic problems in which function evaluations are very costly
Relevance of Discrepancy Optimization

Why not simply use my favorite low discrepancy sequence?

→ tailored for good asymptotic behavior,
  not for our specific setting (e.g., $d=5$, $n=915$)
→ for almost all test cases smaller discrepancy was achieved by our algorithm
  (precisely, 61/62 cases, 31% smaller discrepancy on avg.)
Discrepancy Optimization as Testbed for Genetic Algo.

- Sampling $n$ random points is not too bad
- Better: shuffling, re-evaluating, selection
- Prototype of a GA:

  **Initialization:**
  Choose point sets $X^1, \ldots, X^\lambda$ uniformly at random;
  Compute $d^*_\infty(X^1), \ldots, d^*_\infty(X^\lambda)$;

  **Mutation and Recombination:**
  Create $Y^1, \ldots, Y^\mu$ by mutation and recombination;
  Compute $d^*_\infty(Y^1), \ldots, d^*_\infty(Y^\mu)$;

  **Selection:**
  From $X^1, \ldots, X^\lambda, Y^1, \ldots, Y^\mu$ select $\lambda$ point sets;
  (typically favoring smaller discrepancy)

  **Stop?**
  no → Continue
  yes → Output best configuration
Discrepancy Optimization as Testbed for Genetic Algo.

- Sampling $n$ random points is not too bad
- Better: shuffling, re-evaluating, selection
- Prototype of a GA:

**Initialization:**
- Choose point sets $X^1, \ldots, X^\lambda$ uniformly at random;
- Compute $d^*_\infty(X^1), \ldots, d^*_\infty(X^\lambda)$;

**Mutation and Recombination:**
- Create $Y^1, \ldots, Y^\mu$ by mutation and recombination;
- Compute $d^*_\infty(Y^1), \ldots, d^*_\infty(Y^\mu)$;

**Selection:**
- From $X^1, \ldots, X^\lambda, Y^1, \ldots, Y^\mu$ select $\lambda$ point sets;
  (typically favoring smaller discrepancy)

**Stop?**
- no
- yes

Output best configuration

Doerr, De Rainville: Constructing Low Star Discrepancy Point Sets with Genetic Algorithms
http://qrand.gel.ulaval.ca/
Discrepancy Optimization as Testbed for Genetic Algo.

- Sampling $n$ random points is not too bad
- Better: shuffling, re-evaluating, selection
- Prototype of a GA:

  **Initialization:**
  
  Choose point sets $X^1, \ldots, X^\lambda$ uniformly at random;
  
  Compute $d^*_\infty(X^1), \ldots, d^*_\infty(X^\lambda)$;

  **Mutation and Recombination:**
  
  Create $Y^1, \ldots, Y^\mu$ by mutation and recombination;
  
  Compute $d^*_\infty(Y^1), \ldots, d^*_\infty(Y^\mu)$;

  **Selection:**
  
  From $X^1, \ldots, X^\lambda, Y^1, \ldots, Y^\mu$ select $\lambda$ point sets;
  
  (typically favoring smaller discrepancy)

  - no
  - yes: Output best configuration

**Most problematic part**
Main Challenge: Complexity of Computing $d^*_\infty(X)$

If we are given a point set $X = \{x^1, \ldots, x^n\}$, determining $d^*_\infty(X)$ is difficult:

- Computing star discrepancies is NP-hard
  Gnewuch/Srivastav/Winzen, J. Complexity 2009

- No $n^{o(d)}$ algorithm exists: computation is $W[1]$-hard in $d$
  Giannopoulos/Knauer/Wahlström/Werner, J. Complexity 2012

- Exact computation: $n^{d/2+1}$ algorithm
  Dobkin/Eppstein/Mitchell, ACM Trans. on Graphics 1996

- Several heuristics for proving lower bounds of $d^*_\infty(X)$:
  - Different techniques by E. Thiemard
  - Threshold Accepting-based heuristics
    Winker/Fang, SIAM JoNA 1997
    Gnewuch/Wahlström/Winzen, SIAM JoNA 2012

Main Challenge: Complexity of Computing $d^*_\infty (X)$

If we are given a point set $X = \{x^1, \ldots, x^n\}$, determining $d^*_\infty (X)$ is difficult:

- Computing star discrepancies is NP-hard
  Gnewuch/Srivastav/Winzen, J. Complexity 2009

- No $n^{o(d)}$ algorithm exists: computation is W[1]-hard in $d$
  Giannopoulos/Knauer/Wahlström/Werner, J. Complexity 2012

- Exact computation: $n^{d/2+1}$ algorithm
  Dobkin/Eppstein/Mitchell, ACM Trans. on Graphics 1996

- Several heuristics for proving lower bounds of $d^*_\infty (X)$
  - Different techniques by E. Thiemard
  - Threshold Accepting-based heuristics
    Winker/Fang, SIAM JoNA 1997
    Gnewuch/Wahlström/Winzen, SIAM JoNA 2012


Implementations available on the homepage of Magnus Wahlström.
Discrepancy Optimization as Testbed for Genetic Algo.

- Sampling $n$ random points is not too bad
- Better: shuffling, re-evaluating, selection
- Prototype of a GA:

```
Initialization:
Choose point sets $X^1, \ldots, X^\lambda$ uniformly at random;
Compute $d^*_\infty(X^1), \ldots, d^*_\infty(X^\lambda)$;

Mutation and Recombination:
Create $Y^1, \ldots, Y^\mu$ by mutation and recombination;
Compute $d^*_\infty(Y^1), \ldots, d^*_\infty(Y^\mu)$;

Selection:
From $X^1, \ldots, X^\lambda, Y^1, \ldots, Y^\mu$ select $\lambda$ point sets;
(typically favoring smaller discrepancy)
```

Done either by the exact DEM-algo. or by the TA-based heuristic of GWW

Stop?  

yes  
Output best configuration

No
Discrepancy Optimization as Testbed for Genetic Algo.

- Sampling $n$ random points is not too bad
- Better: shuffling, re-evaluating, selection
- Prototype of a GA:

  **Initialization:**
  Choose point sets $X^1, ..., X^\lambda$ uniformly at random;
  Compute $d^*_\infty(X^1), ..., d^*_\infty(X^\lambda)$;

  **Mutation and Recombination:**
  Create $Y^1, ..., Y^\mu$ by mutation and recombination;
  Compute $d^*_\infty(Y^1), ..., d^*_\infty(Y^\mu)$;

  **Selection:**
  From $X^1, ..., X^\lambda, Y^1, ..., Y^\mu$ select $\lambda$ point sets;
  (typically favoring smaller discrepancy)

  **Stop?**
  yes: Output best configuration
  no: Return to Optimization

Want to sample only low discrepancy point sets

Doerr, De Rainville: Constructing Low Star Discrepancy Point Sets with Genetic Algorithms  
http://qrand.gel.ulaval.ca/
Our Sample Space for Low Discrepancy Point Sets

We optimize generalized Halton point sets

**Halton**

\[ \varphi_p(i) := \sum_{\ell=1}^{k} d_\ell p^{-\ell}, \]

where \( i = d_k d_{k-1} \ldots d_1 \) is the digital expansion of \( i \) in base \( p \); i.e., \( i = \sum_{\ell=1}^{k} d_\ell p^{\ell-1} \)

\( \pi_p \) is a permutation of \( \{0, 1, \ldots, p-1\} \) with fixpoint \( \pi_p(0) = 0 \) (so that \( \varphi_p^{\pi_p}(i) \neq 0 \)).

\[ x(i) := (\varphi_{p_1}(i), \ldots, \varphi_{p_d}(i)) \]

**Generalized Halton**

\[ \varphi_{p}^{\pi_p}(i) := \sum_{\ell=1}^{k} \pi_p(d_\ell) p^{-\ell}, \]

\[ x^{(i)}(\Pi) := (\varphi_{p_1}^{\pi_{p_1}}(i), \ldots, \varphi_{p_d}^{\pi_{p_d}}(i)) \]

We optimize over the permutations \( \Pi = (\pi_{p_1}, \ldots, \pi_{p_d}) \). (for \( \pi_{p_k} \) there are \((p_k - 1)!\) possibilities)
Brief Sketch of the Algorithm

- We optimize generalized Halton point sets
- We use a standard genetic algorithm
  (no fine tuning, no parameter testing, just a plain GA)
**Brief Sketch of the Algorithm**

Sample $\Pi_1, \ldots, \Pi_\lambda$ uniformly at random;
Compute $d^*_\infty (X(\Pi_1)), \ldots, d^*_\infty (X(\Pi_\lambda))$;

For $i = 1, \ldots, \mu$ do
- mutate a randomly chosen $\Pi_i$ or
- recombine randomly selected $\Pi_i, \Pi_i'$, and compute $d^*_\infty (X(\widehat{\Pi}_i))$;

From $\Pi_1, \ldots, \Pi_\lambda, \widehat{\Pi}_1, \ldots, \widehat{\Pi}_\mu$ select $\lambda$ permutations (giving preference to smaller discrepancy)

Stop? Output $\Pi_i$;
Brief Sketch of the Algorithm

Sample $\Pi_1, \ldots, \Pi_\lambda$ uniformly at random; Compute $d^*_\infty (X(\Pi_1)), \ldots, d^*_\infty (X(\Pi_\lambda))$;

For $i = 1, \ldots, \mu$ do
- mutate a randomly chosen $\Pi_i$ or
- recombine randomly selected $\Pi_i, \Pi_i$,

and compute $d^*_\infty (X(\Pi_i))$;

Re-evaluate $d^*_\infty (X(\Pi_1)), \ldots, d^*_\infty (X(\Pi_\lambda))$;

From $\Pi_1, \ldots, \Pi_\lambda, \overline{\Pi}_1, \ldots, \overline{\Pi}_\mu$ select $\lambda$ permutations (giving preference to smaller discrepancy);

Do a critical re-evaluation of $X(\Pi_i)$ and output best $\Pi_i$
Results

- 61/62 cases: smaller discrepancy than literature
- In average, 36% improvement
- Comparison with Doerr/Gnewuch/Wahlström 2010: (exact DEM-algorithm feasible)

<table>
<thead>
<tr>
<th>$d$</th>
<th>$n$</th>
<th>Results from [DGW10]</th>
<th>Optimized Halton</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>95</td>
<td>~ 0.11</td>
<td>0.08445</td>
<td>23%</td>
</tr>
<tr>
<td>7</td>
<td>65</td>
<td>0.150</td>
<td>0.1361</td>
<td>9%</td>
</tr>
<tr>
<td>7</td>
<td>145</td>
<td>0.098</td>
<td>0.08640</td>
<td>12%</td>
</tr>
<tr>
<td>9</td>
<td>85</td>
<td>0.170</td>
<td>0.1435</td>
<td>16%</td>
</tr>
</tbody>
</table>

Table 2: Exact discrepancy results for the sequences presented in [DGW10].
Results

- 61/62 cases: smaller discrepancy than literature
- In average, 36% improvement
- Comparison with Doerr/Gnewuch/Wahlström 2010: (lower bounds, exact DEM-algorithm not feasible)

<table>
<thead>
<tr>
<th>$d$</th>
<th>$n$</th>
<th>Results from [DGW10]</th>
<th>Optimized Halton</th>
</tr>
</thead>
<tbody>
<tr>
<td>9*</td>
<td>145</td>
<td>0.119</td>
<td>0.1083 9%</td>
</tr>
<tr>
<td>12*</td>
<td>65</td>
<td>0.276</td>
<td>0.2049 26%</td>
</tr>
<tr>
<td>12</td>
<td>145</td>
<td>0.156</td>
<td>0.1354 13%</td>
</tr>
<tr>
<td>15</td>
<td>65</td>
<td>0.322</td>
<td>0.2413 25%</td>
</tr>
<tr>
<td>15</td>
<td>95</td>
<td>0.258</td>
<td>0.1969 24%</td>
</tr>
<tr>
<td>15</td>
<td>145</td>
<td>0.198</td>
<td>0.1589 20%</td>
</tr>
<tr>
<td>18</td>
<td>95</td>
<td>0.293</td>
<td>0.2237 24%</td>
</tr>
<tr>
<td>18</td>
<td>145</td>
<td>0.230</td>
<td>0.1823 21%</td>
</tr>
<tr>
<td>20</td>
<td>145</td>
<td>0.239</td>
<td>0.1947 19%</td>
</tr>
<tr>
<td>21</td>
<td>95</td>
<td>0.299</td>
<td>0.2434 19%</td>
</tr>
</tbody>
</table>

Table 3: Approximated discrepancy results for the sequences presented in [DGW10]. Lines marked with a star indicate that our final discrepancy measure is exact.
Results

- 61/62 cases: smaller discrepancy than literature
- In average, 36% improvement
- Comparison with Tiemard 2011:
  (exact evaluation with DEM-algorithm)

<table>
<thead>
<tr>
<th>$d$</th>
<th>$n$</th>
<th>Results from [Thi01a, Thi01b]</th>
<th>Optimized Halton</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>125</td>
<td>0.089387</td>
<td>0.05609 37%</td>
</tr>
<tr>
<td>4</td>
<td>615</td>
<td>0.017725</td>
<td>0.01905 -7%</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>0.238297</td>
<td>0.18 24%</td>
</tr>
<tr>
<td>5</td>
<td>125</td>
<td>0.141788</td>
<td>0.07158 50%</td>
</tr>
<tr>
<td>5</td>
<td>625</td>
<td>0.266623</td>
<td>0.2352 12%</td>
</tr>
<tr>
<td>6</td>
<td>49</td>
<td>0.210972</td>
<td>0.13959 34%</td>
</tr>
<tr>
<td>6</td>
<td>343</td>
<td>0.089884</td>
<td>0.04547 49%</td>
</tr>
<tr>
<td>7</td>
<td>49</td>
<td>0.269011</td>
<td>0.1641 39%</td>
</tr>
<tr>
<td>8</td>
<td>121</td>
<td>0.170184</td>
<td>0.109 36%</td>
</tr>
<tr>
<td>9</td>
<td>121</td>
<td>0.212126</td>
<td>0.1244 41%</td>
</tr>
</tbody>
</table>

Table 4: Exact discrepancy results for the sequences presented in [Thi01a, Thi01b].

Doerr, De Rainville: Constructing Low Star Discrepancy Point Sets with Genetic Algorithms
http://qrand.gel.ulaval.ca/
Results

- 61/62 cases: smaller discrepancy than literature
- In average, 36% improvement
- Comparison with Thiemard 2011:
  (lower bounds, exact DEM-algorithm not feasible)

<table>
<thead>
<tr>
<th>$d$</th>
<th>$n$</th>
<th>Results from [Thi01a, Thi01b]</th>
<th>Optimized Halton</th>
</tr>
</thead>
<tbody>
<tr>
<td>7*</td>
<td>343</td>
<td>0.129832</td>
<td>0.05192    60%</td>
</tr>
<tr>
<td>7</td>
<td>2401</td>
<td>0.030518</td>
<td>0.01518    50%</td>
</tr>
<tr>
<td>10*</td>
<td>121</td>
<td>0.2574323</td>
<td>0.1334     48%</td>
</tr>
<tr>
<td>10</td>
<td>1331</td>
<td>0.093028</td>
<td>0.03251    65%</td>
</tr>
<tr>
<td>11*</td>
<td>121</td>
<td>0.301048</td>
<td>0.1402     53%</td>
</tr>
<tr>
<td>12</td>
<td>169</td>
<td>0.271837</td>
<td>0.1211     55%</td>
</tr>
<tr>
<td>12</td>
<td>2197</td>
<td>0.096713</td>
<td>0.02857    70%</td>
</tr>
<tr>
<td>15</td>
<td>289</td>
<td>0.256021</td>
<td>0.1083     58%</td>
</tr>
<tr>
<td>15</td>
<td>4913</td>
<td>0.085855</td>
<td>0.02239    74%</td>
</tr>
<tr>
<td>20</td>
<td>529</td>
<td>0.259366</td>
<td>0.09859    62%</td>
</tr>
<tr>
<td>100</td>
<td>101</td>
<td>0.954159</td>
<td>0.5458     43%</td>
</tr>
</tbody>
</table>

Table 5: Approximated discrepancy results for the sequences presented in [Thi01a, Thi01b]. Lines marked with a star indicate that our final discrepancy measure is exact.
Results

Figure 4: Exact discrepancy results on 7 dimension point sets.
Results

Figure 5: Best exact star discrepancy values in $d = 4$ to $d = 6$. 

Doerr, De Rainville: Constructing Low Star Discrepancy Point Sets with Genetic Algorithms  
http://qrand.gel.ulaval.ca/
Results

Figure 6: Best approximated star discrepancy values in $d = 10$ to $d = 20$. 
Inverse Star Discrepancies

- Not much empirical work done
- Novak/Wozniakowski 2010:
  - explicitly find point sets of discrepancy $\leq 1/4$
  - $d=15$, $n=1,528$
  - $d=30$, $n=3,187$
  - $d=50$, $n=5,517$
  
  (existence follows from B.Doerr/Gnewuch/Srivastav 2005)
- Hinrichs 2012: “lifting technique”

<table>
<thead>
<tr>
<th></th>
<th>0.25</th>
<th>0.125</th>
<th>0.0575</th>
<th>0.0136</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>1528</td>
<td>8</td>
<td>764</td>
<td>4</td>
</tr>
<tr>
<td>30</td>
<td>3187</td>
<td>15</td>
<td>1593</td>
<td>8</td>
</tr>
<tr>
<td>50</td>
<td>5517</td>
<td>25</td>
<td>2758</td>
<td>13</td>
</tr>
</tbody>
</table>
## Inverse Star Discrepancies

Problem from Novak/Wozniakowski 2010 and Hinrichs 2012

<table>
<thead>
<tr>
<th></th>
<th>0.25</th>
<th>0.125</th>
<th>0.0575</th>
<th>0.0136</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>1528</td>
<td>8</td>
<td>764</td>
<td>104</td>
</tr>
<tr>
<td>30</td>
<td>3187</td>
<td>15</td>
<td>1593</td>
<td>251*</td>
</tr>
<tr>
<td>50</td>
<td>5517</td>
<td>25</td>
<td>2758</td>
<td>513*</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>764</td>
<td>172</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1593</td>
<td>537*</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2758</td>
<td>1239*</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>764</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8</td>
<td>1593</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>13</td>
<td>2758</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1048</td>
<td>&gt;3000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&gt;1593</td>
<td>&gt;5000</td>
</tr>
</tbody>
</table>

*: exact DEM evaluation infeasible. Verification needed.
Evaluation of the Algorithm

✓ Fairly simple, easy to implement
✓ Very promising results, even w/o finetuning
✓ very short running times → “quick & dirty” answers possible

Problems:

- no guarantee to find low discrepancy point sets
- intermediate TA-based evaluation can be misleading
- exact evaluation remains infeasible for large dimensions
- currently works only for generalized Halton sequences
Summary & Future Work

- We have presented an algorithm that, seemingly, computes quite good low discrepancy point sets.
- Long term goal: provide a database with these point sets and algorithms.

Future Work:

- Existing techniques: tuning of the algorithm, test other point sets, comparison of point sequences in low & medium dimensions.
- Low discrepancy point sets in dimensions $\geq 10$
  - Exact discrepancy computation remains infeasible, but can we compute reasonably good upper bounds for $d_{\infty}^*(X)$?
  - Better medium/high-dimensional point sets through Hinrich’s “lifting technique”?
- Spreading the word: how to reach the (applied) communities?
- Other applications?
References 1/4

- P. Bundschuh, Y. Zhu
  *A method for exact calculation of the discrepancy of low-dimensional point sets (I)*
  In: Abhandlungen aus dem Mathematischen Seminar der Univ. Hamburg, 63(1):115--133
  Springer, 1993

- L. De Clerck
  *A method for exact calculation of the star-discrepancy of plane sets applied to the sequence of Hammersley*
  In: Monatshefte für Mathematik, 101(4):261--278
  Springer, 1986

- Carola Doerr, François-Michel De Rainville
  *Constructing low star discrepancy point sets with genetic algorithms*
  In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO ’13
  Association for Computing Machinery (ACM), 2013. To appear

- David P. Dobkin, David Eppstein, Don P. Mitchell
  *Computing the discrepancy with applications to supersampling patterns*
  Association for Computing Machinery (ACM), 1996
References 2/4

- Benjamin Doerr, Michael Gnewuch, Magnus Wahlström
  Algorithmic construction of low discrepancy point sets via dependent randomized rounding
  In: Journal of Complexity, 26:490--507
  Elsevier, 2010

- Panos Giannopoulos, Christian Knauer, Magnus Wahlström, Daniel Werner
  Hardness of discrepancy computation and epsilon-net verification in high dimension
  In: Journal of Complexity, 28(2):162--176
  Elsevier, 2012

- Michael Gnewuch, Anand Srivastav, Carola Winzen
  Finding optimal volume subintervals with k points and calculating the star discrepancy are NP-hard problems
  In: Journal of Complexity, 25(2):115--127
  Elsevier, 2009

- Michael Gnewuch, Magnus Wahlström, Carola Winzen
  A new randomized algorithm to approximate the star discrepancy based on threshold accepting
  In: SIAM Journal on Numerical Analysis, 50:781--807
  Society for Industrial and Applied Mathematics (SIAM), 2012
References 3/4

- Harald Niederreiter
  *Methods for estimating discrepancy*
  In: Applications of Number Theory to Numerical Analysis, 203--236
  Academic Press, 1972

- Erich Novak and Henryk Wozniakowski
  *Standard Information for Functionals*
  In: Tractability of Multivariate Problems, vol. 2
  EMS Tracts in Mathematics, European Mathematical Society, 2010

- François-Michel De Rainville, Christian Gagné, Olivier Teytaud, Denis Laurendeau
  *Evolutionary optimization of low-discrepancy sequences*
  Association for Computing Machinery (ACM), 2012

- Manan Shah
  *A genetic algorithm approach to estimate lower bounds of the star discrepancy*
  In: Monte Carlo Methods and Applications, 16(3-4):379--398
  De Gruyter, 2010
References 4/4

- Eric Thiémard
  Computing bounds for the star discrepancy
  In: Computing 65(2):169--186
  Springer, 2000

- Eric Thiémard
  An algorithm to compute bounds for the star discrepancy
  In: Journal of Complexity, 17(4):850--880
  Elsevier, 2001

- Eric Thiémard
  Optimal volume subintervals with k points and star discrepancy via integer programming
  In: Mathematical Methods of Operations Research, 54(1):21--45
  Springer, 2001

- Peter Winker and Kai-Tai Fang
  Applications of threshold-accepting to the evaluation of the discrepancy of a set of points
  In: SIAM Journal on Numerical Analysis, 34:2028--2042
  Society for Industrial and Applied Mathematics (SIAM), 1997