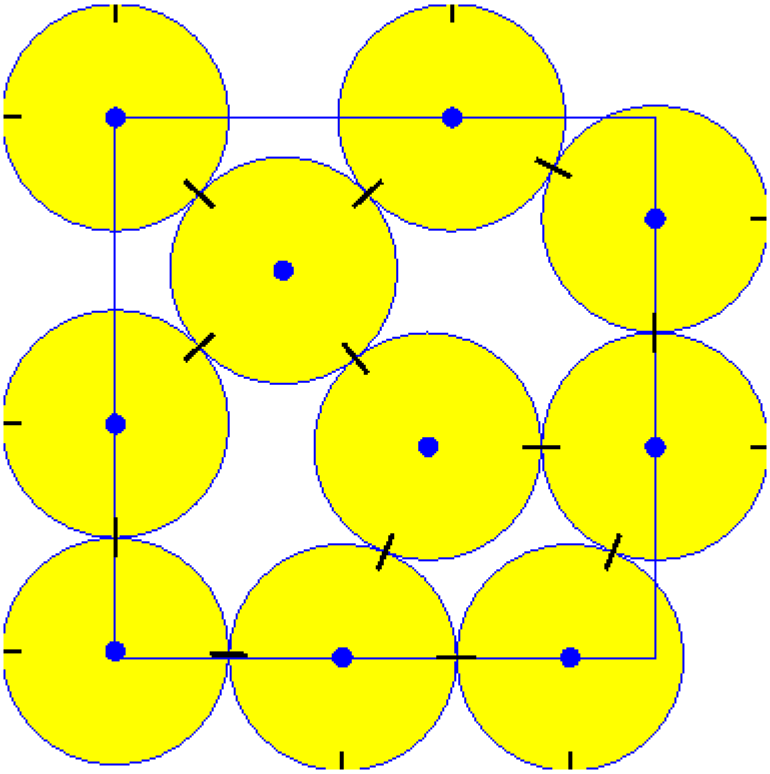
# Improved Methods for Solving "Packing of Equal Circles into the Unit Square" Problems

Mihály Csaba Markót
University of Szeged, Hungary



Dagstuhl, January, 2003.

## A possible approach to the 'Circle Packing' problem

**A1. Place $n$ points into the unit square maximizing the minimal distance between the points.**

This approach leads to the global optimization problem

$$\max \ \left( \min_{1 \le i < j \le n} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \right),$$

$$\text{subject to} \qquad 0 \le x_i, y_i \le 1, \qquad i = 1, 2, \ldots, n.$$

The objective function to be maximized is the following:

$$f : \mathbb{R}^{2n} \to \mathbb{R} \ ,$$

$$f(\mathbf{x}, \mathbf{y}) = \min_{1 \le i < j \le n} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} = \min_{1 \le i < j \le n} d_{ij}. \tag{1}$$

## Equivalent approaches

**A2. Place $n$ non-overlapping equal circles with maximal radius into the unit square.**
**A3. Find the smallest square into which it is possible to place $n$ points in such a way that the distances between the points are at least 1.**
**A4. Find the smallest square into which it is possible to place $n$ non-overlapping equal circles with unit radius.**

Generally approach A1 is preferred since it results in a **bound-constrained** problem instead of the **equality/inequality constrained problems** provided by the other approaches.

## Research directions dealing with packing problems

**“By-hand” methods**

- Give theoretical proof for the optimality of a packing for a given $n$
- Give lower and upper bounds for the optimal solution in theoretical way

**“Computer-aided proofs” – result-checking is vital**

- Compute the value of the optimal solution for a given $n$
- Compute numerical lower and upper bounds for the optimal solution

## Brief history

- First appearance in the mathematical literature – L. Moser (1960)
- Optimality for $n = 2, 3, 4, 5, 6, 7, 8, 9$ in theoretical way – J. Schaer, A. Meir (1965)
- Optimality for $n = 14, 16, 25, 36$ in theoretical way – K. Kirchner, G. Wengerodt (1983,1987)
- Optimality up to 20 circles with a computer-aided method – R. Peikert, D. Würtz, M. Monagan, C. de Groot (1990, 1992)
- “Good packings” (without proving optimality)

  Up to 30 – C.D. Maranas, C.A. Floudas, P.M. Pardalos (1995)

  Up to 52 and $n = 54, 56, 60, 61, 62, 72, 78$ – R.L. Graham and B.D. Lubachevsky (1996)
- Optimality up to 27 with a computer-aided technique based on Peikert et al.’s one – K. Nurmela and P.R.J. Östergård (1999)
- “Good packings” up to 100 with the TAMSASS-PECS algorithm – L.G. Casado, I. García, P.G. Szabó and T. Csendes (1999)
- A fully interval-arithmetic based algorithm for local validation of optimal numerical solutions – M.Cs. Markót (2000)
- Nowadays: several new approaches and attempts to improve the current best packings and to prove optimality for $n > 27$

## Interval Branch-and-Bound algorithm

`Global Optimize`$(f, X_0, \varepsilon, Maximum, ResultList)$

1. $Z := X_0$; `WorkTree` $:= \{\}$; $\tilde{f} = \underline{F}(Z)$;
2. `Bisection`$(Z, U^1, U^2)$;
3. **For** $i := 1$ **to** 2 **do**

3.1. `Try to improve` $\tilde{f}$;

3.2. `Accelerating devices for` $U^i$;

3.3. **If** (`$U^i$ can be deleted as a whole`) **then next** `i`;

3.4. **Else** `Insert(WorkTree,`$(U^i, \overline{F}(U^i))$`)`;

4. **While** (`WorkTree is not empty`) **do**

4.1. $(Z, \overline{F}(Z))$ := `Head(WorkTree)`;

4.2. `Delete(Head(WorkTree))`;

4.3. **If** $(w(F(Z)) < \varepsilon)$ **then**

4.3.1. `Insert(ResultList,`$(Z, \overline{F}(Z))$`)`;

4.3.2. `Maximum` $:= (\tilde{f}, \overline{F}(Z))$;

4.3.3. **Return to While**

4.4. **Else go to** 2.

5. **Return** (`ResultList,Maximum`);

# An interval inclusion function for the 'Packing Circles' problem

**Theorem** (Markót, 2000): Let $(\mathbf{X}, \mathbf{Y}) \subseteq [0,1]^{2n}$, and

$$D_{ij} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}, \qquad \text{for} \;\; 1 \le i < j \le n,$$

and let

$$a = \min_{1 \le i < j \le n} \underline{D_{ij}} \qquad a \in I\!R,$$
$$b = \min_{1 \le i < j \le n} \overline{D_{ij}} \qquad b \in I\!R.$$

Then the function $F(\mathbf{X}, \mathbf{Y}) = [\, a, b \,]$ is an inclusion function of $f(\mathbf{x}, \mathbf{y})$ in the $(\mathbf{X}, \mathbf{Y})$ $2n$-dimensional interval.

**Example:** Compute $F(\mathbf{X}, \mathbf{Y})$ for the interval box $X_1 = [0.0, 0.1]$, $X_2 = [0.5, 0.6]$, $X_3 = [0.6, 0.7]$, $Y_1 = [0.2, 0.7]$, $Y_2 = [0.7, 1.0]$, $Y_3 = [0.1, 0.2]$.

$$\begin{aligned}
{D_{12}}^2 &= ([0.0, 0.1] - [0.5, 0.6])^2 + ([0.2, 0.7] - [0.7, 1.0])^2 = [0.16, 1.0] \;, \\
{D_{13}}^2 &= ([0.0, 0.1] - [0.6, 0.7])^2 + ([0.2, 0.7] - [0.1, 0.2])^2 = [0.25, 0.85] \;, \\
{D_{23}}^2 &= ([0.5, 0.6] - [0.6, 0.7])^2 + ([0.7, 1.0] - [0.1, 0.2])^2 = [0.25, 0.85] \;.
\end{aligned}$$

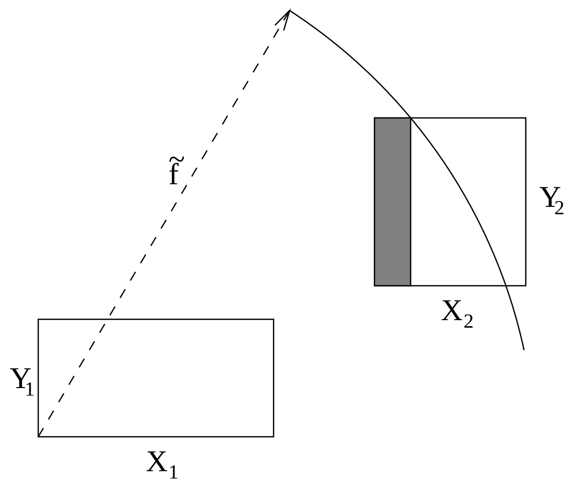
After finding the minimal lower and upper bounds we obtain

$$a = \sqrt{0.16} = 0.4 \;, \; b = \sqrt{0.85} \;,$$

which results in the interval inclusion $F(\mathbf{X}, \mathbf{Y}) = [0.4, \sqrt{0.85}]$.

# Method of active areas

Assume that a lower bound $\tilde{f}$ for the maximum value is given. Then from every $(X_j, Y_j)$ pair (i.e. a rectangle in the unit square) one can delete those points for which the distance from all the points of an other $(X_i, Y_i)$ rectangle is less than $\tilde{f}$.




`Reducing the active area by pruning rectangle` $(X_2, Y_2)$

**Vital:** prune a given rectangle using *all the other rectangles at the same time*. This requires the approximation of the exact remaining active areas.

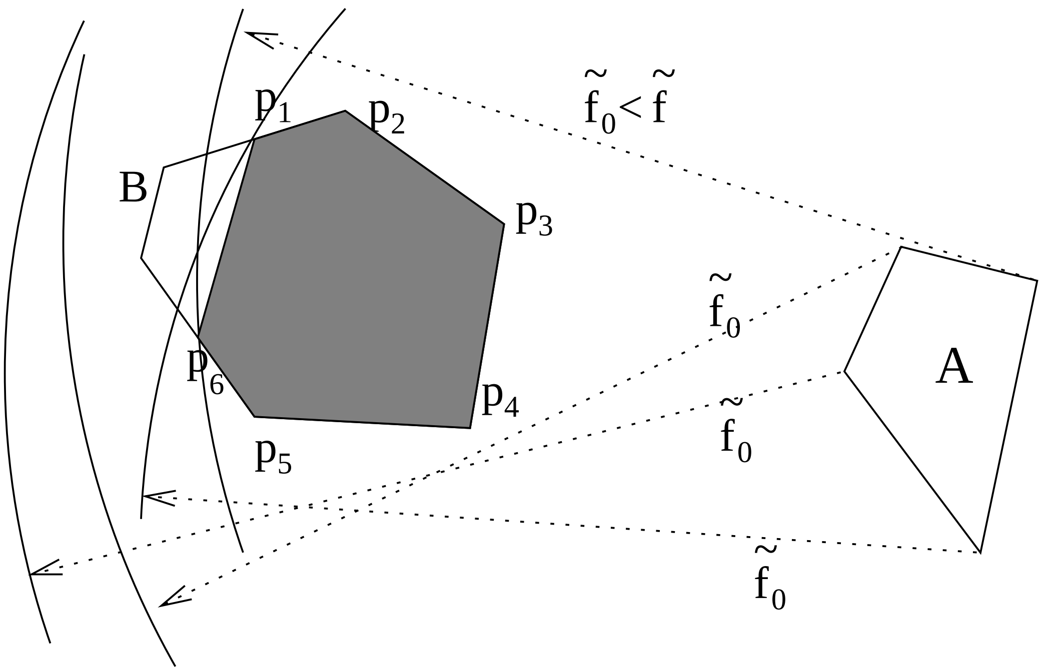
Some possible ways are:

- **Splitting the rectangles into $k \times l$ cells** (C. de Groot, et al.)
  - Advantages: $k$ and $l$ can dynamically be changed; easy to implement with interval methods.
  - Disadvantage: rough approximation $\rightarrow$ inefficient as compared to the polygon approximation.
- **Approximating the active areas by polygons** (K. J. Nurmela, et al.)
  - Advantage: more precise approximation;
  - Disadvantage: hard to implement in a reliable way.

# Method of active areas (cont.)

**Lemma** (Nurmela and Östergård, 1999): If a point $p$ is at distance less than $\tilde{f}$ from the vertices of a polygon $A$, it is at distance less than $\tilde{f}$ from all points of $A$.

**Theorem** (Nurmela and Östergård, 1999): Assume that $p_1, p_2, \ldots, p_k$ are distinct points on the boundary of a polygon $B$, that $p_i p_{i+1}$ for $2 \leq i \leq k-2$ are edges of $B$, and that $p_1 p_2$ and $p_{k-1} p_k$ lie on the edges of $B$. If the points $p_i$, $1 \leq i \leq k$ are at distance less than $\tilde{f}$ from all vertices of $A$, then the points in the polygon formed by $p_1, p_2, \ldots, p_k$ is at distance less than $\tilde{f}$ from all points in $A$.




`Method of active areas using polygons (`$k = 6$`)`

# Method of active areas - the interval approach

- Representation of polygons: a sequence of consecutive vertices, where the coordinates of the vertices are machine numbers.

**- The main steps of the interval-based algorithm:**

- Determine a set of consecutive vertices $p_2, \ldots p_{k-1}$ which are **guaranteed** to be at distance less than $\tilde{f}$ from all vertices of $A$. This may be a subsequence of that of the $\{p_i\}$ sequence resulted by exact computations, but the Theorem above remains valid.
- Determine the rectangular inclusions $P_1$ and $P_k$ of the points $p_1$ and $p_k$, respectively.
- Compute a polygon including all the possible remaining polygons resulted in by all the possible edges $p'_1 p'_k$ (where $p'_1 \in P_1$, $p'_2 \in P_2$).

- Each elimination step performed for $B$ results in either

- an empty polygon,
- the original polygon $B$, or
- a polygon containing the polygon would be resulted in assuming exact arithmetic.

# Obtaining global solutions

**Tiling methods** ( de Groot et al., 1990 ): Assuming that a lower bound $\tilde{f}$ for the maximum value is given, split the unit square into regions (tiles) in such a way, that the distance between any two points in each tile is less than $\tilde{f}$. Then for a packing having objective function value greater than or equal to $\tilde{f}$ each tile can contain at most one point.

- The optimality of a given packing is proved by searching on tile combinations.
- Symmetric properties can be used to reduce the number of tile combinations.
- If all the tile combinations are eliminated using a hyphotethic bound instead of the best known lower bound, then the hyphotethic bound can be considered as an **upper bound** of the current problem.
- When splitting the unit square into $k \times l$ identical rectangles, the minimal number of the initial tile combinations is

$$\min\left\{ \tbinom{k\cdot l}{n} \ \middle|\ k, l \geq 1 \text{ integers}, \ \sqrt{\frac{1}{k^2} + \frac{1}{l^2}} < \tilde{f} \right\}.$$

- In the previous studies the initial tile combinations are eliminated sequentially one after the other; for $n = 27$ the number of initial tile combinations is $\approx 9.4 \cdot 10^7$ (this case has been solved in about one month of CPU time). For $n = 28$ this number is $\approx 5.3 \cdot 10^{10}$.

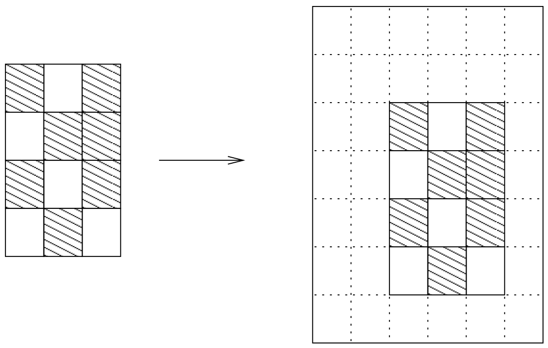
# A new approach for eliminating tile combinations

- Instead of processing the combinations sequentially, use local information to erase groups of combinations:

- If a set of tiles (a pattern) can fully be erased, then we can erase all the tile combinations which contain this set.

- If the square is divided horizontally and vertically into **equal** rectangular pieces, and a pattern can partly be erased, then the remaining area of the pattern can be transformed (shifted or rotated) into the position of another sets of tiles to determine other remaining areas.

- For $n = 28$ a $7 \times 6$ splitting can be applied.



`Applying tile sub-combinations`

| | | | | | |
|---|---|---|---|---|---|
| 7 | 14 | 21 | 28 | 35 | 42 |
| 6 | 13 | 20 | 27 | 34 | 41 |
| 5 | 12 | 19 | 26 | 33 | 40 |
| 4 | 11 | 18 | 25 | 32 | 39 |
| 3 | 10 | 17 | 24 | 31 | 38 |
| 2 | 9 | 16 | 23 | 30 | 37 |
| 1 | 8 | 15 | 22 | 29 | 36 |

`Tile labels for` $n = 28$

# Optimality proof for $n = 28$

- Eliminate tile sets in intermediate phases: run at most 3 iteration steps of the B&B algorithm, store the possible remaining areas.
- At the beginning of each phase determine some sets of tile combinations consisting either of the remaining areas of the previous phases (or its transformations) or of full tiles:

  $D^m_{s..f} := \{$combinations where $m$ tiles is chosen from the fields $s, \ldots, f\}$.
- Process the elements of $D^m_{s..f}$ sequentially by the optimization algorithm using $\tilde{f}$. Result sets: $\bar{D}^m_{s..f}$.
- Goal: determine $\bar{D}^{28}_{1..42}$.

**Main steps of the procedure:**

Phase 1: Determine $\bar{D}^*_{1..21}$.

Phase 2: Determine $\bar{D}^*_{1..28}$ from $\bar{D}^*_{1..21}$ and from $\bar{D}^*_{8..28}$.

Phase 3: Determine $\bar{D}^{28}_{1..42}$ from $\bar{D}^*_{1..28}$ and from $\bar{D}^*_{15..42}$.

Phase 4: Refine $\bar{D}^{28}_{1..42}$ by applying at most 3000 iteration steps of the B&B algorithm for each element of the result set of Phase 3.

**Acceleration steps:**

- $\bar{D}^{16}_{22..42} := \bar{D}^{16}_{1..21}$ proved to be empty $\rightarrow$ no need to compute $\bar{D}^m_{1..21}$ for $m \geq 17$.
- $\bar{D}^{12}_{29..42} := \bar{D}^{12}_{1..14}$ proved to be empty $\rightarrow$ no need to compute $\bar{D}^m_{1..28}$ for $m \leq 16$.
- Filtering symmetric cases.

# Optimality for $n = 28$, computational results

**Hardware :** PC with a Pentium IV 1800 MHz processor and with 1 GB RAM.

**Software:** Linux operation system, GNU C/C++ compiler, C–XSC Toolbox and PROFIL/BIAS interval libraries.

| $D^m_{s..f}$ | $\lvert D^m_{s..f}\rvert$ | CPU (sec) | NFE | NMAA | $\bar{D}^m_{s..f}$ | $\lvert\bar{D}^m_{s..f}\rvert$ |
|---|---|---|---|---|---|---|
| | | | **Phase 1** | | | |
| $D^{14}_{1..21}$ | 116 280 | 788 | 54 849 | 60 253 | $\bar{D}^{14}_{1..21}$ | 17 803 |
| $D^{15}_{1..21}$ | 54 264 | 147 | 4 393 | 16 652 | $\bar{D}^{15}_{1..21}$ | 1 078 |
| $D^{13}_{1..21}$ | 203 490 | 2 576 | 232 254 | 174 998 | $\bar{D}^{13}_{1..21}$ | 77 840 |
| $D^{16}_{1..21}$ | 20 349 | 22 | 27 | 5 253 | $\bar{D}^{16}_{1..21}$ | 0 |
| | | | **Phase 2** | | | |
| $D^{17}_{1..28}$ | 1 048 486 | 104 060 | 7 465 223 | 4 969 787 | $\bar{D}^{17}_{1..28}$ | 629 484 |
| $D^{18}_{1..28}$ | 1 013 184 | 64 790 | 4 214 161 | 3 365 831 | $\bar{D}^{18}_{1..28}$ | 334 370 |
| $D^{19}_{1..28}$ | 146 305 | 4 917 | 211 578 | 280 655 | $\bar{D}^{19}_{1..28}$ | 13 246 |
| $D^{20}_{1..28}$ | 5 394 | 36 | 540 | 5 749 | $\bar{D}^{20}_{1..28}$ | 31 |
| $D^{21}_{1..28}$ | 0 | - | - | - | $\bar{D}^{21}_{1..28}$ | 0 |
| $D^{22}_{1..28}$ | 0 | - | - | - | $\bar{D}^{22}_{1..28}$ | 0 |
| | | | **Phase 3** | | | |
| $D^{28}_{1..42}$ | 1 212 766 | 10 137 | 45 696 | 1 248 939 | $\bar{D}^{28}_{1..42}$ | 564 |
| | | | **Phase 4** | | | |
| | | 1 345 | 68 254 | 41 956 | $\bar{D}^{28}_{1..42}$ | 6 |
| **Summary** | | 188 818 | 12 296 975 | 10 170 073 | | |

Computational details. The sets $D^m_{s..f}$ are processed by the global optimization algorithm to get $\bar{D}^m_{s..f}$.

- Total running time (including tile set transformations): 52 hours.
- The 6 elements of the result set of Phase 4 are proved to come from symmetric cases:
  - → it is enough to consider one of them to give tighter enclosure for the global maximizer/maximum.
  - → the **structure** of the optimal packing is unique.

# Optimal packing for $n = 28$

The currently known best enclosure for one of the global maximizers:

$$
\begin{array}{lcl}
[0.0000000000000000, 0.0000000000344040] & , & [0.0000000000000000, 0.000000000441672] \\
[0.0000000000000000, 0.0000000002314576] & , & [0.2833357017046658, 0.2833357020022985] \\
[0.0000000000000000, 0.0000000002307674] & , & [0.5138711953469158, 0.5138711956445486] \\
[0.0000000000000000, 0.0000000001977346] & , & [0.7444066889891658, 0.7444066892860408] \\
[0.0000000000000000, 0.0000000000211772] & , & [0.9999999999682048, 1.0000000000000000] \\
[0.1818703764267072, 0.1818703765426157] & , & [0.1416678508523484, 0.1416678510229847] \\
[0.1996495938823974, 0.1996495949826877] & , & [0.3986034479423025, 0.3986034488096029] \\
[0.1996495938824742, 0.1996495945360366] & , & [0.6291389415845525, 0.6291389424513962] \\
[0.1918713857522944, 0.1918713858546747] & , & [0.8722033444787086, 0.8722033446430095] \\
[0.3637407528534142, 0.3637407567732359] & , & [0.0000000000000000, 0.0000000035110862] \\
[0.3815199703962453, 0.3815199708329341] & , & [0.2569355970663725, 0.2569355978420794] \\
[0.3992991877579223, 0.4023815123711082] & , & [0.5089496079216517, 0.5138711956474279] \\
[0.3915209796974742, 0.3915209798150105] & , & [0.7569355976188575, 0.7569355978188969] \\
[0.3837427715045880, 0.3837427717022675] & , & [0.9999999998410170, 1.0000000000000000] \\
[0.5633903468231437, 0.5633903500005439] & , & [0.1152677458213779, 0.1152677508698651] \\
[0.5839312816067412, 0.5839312817378985] & , & [0.3672817571397007, 0.3672817573495594] \\
[0.5911705737019623, 0.5911705737771878] & , & [0.6416678508754032, 0.6416678509873931] \\
[0.5833923654509524, 0.5833923656555152] & , & [0.8847322529831440, 0.8847322533243109] \\
[0.7630399407924907, 0.7630399423249100] & , & [0.0000000000000000, 0.0000000003605143] \\
[0.7694645060939898, 0.7694645063577499] & , & [0.2304459563241970, 0.2304459566891774] \\
[0.7727203431190995, 0.7727203431363298] & , & [0.4995893688680368, 0.4995893689055492] \\
[0.7830419594696230, 0.7830419596088718] & , & [0.7694645062219400, 0.7694645063577497] \\
[0.7830419593949475, 0.7830419596088718] & , & [0.9999999998641899, 1.0000000000000000] \\
[0.9935754344347408, 0.9935754363327754] & , & [0.0000000000000000, 0.0000000000799076] \\
[0.9999999997362400, 1.0000000000000000] & , & [0.2304459563241970, 0.2304459564036267] \\
[0.9999999999865099, 1.0000000000000000] & , & [0.4609814499664471, 0.4609814500458598] \\
[0.9999999998607514, 1.0000000000000000] & , & [0.6915169436086973, 0.6915169439962785] \\
[0.9999999997876501, 1.0000000000000000] & , & [0.9220524372509473, 0.9220524378463795]
\end{array}
$$

The currently known best enclosure for the global maximum value:

$$F^* = [\underline{0.2305354936}422503, \underline{0.2305354936}592372]$$

$$w(F^*) \approx 1.7 \cdot 10^{-11}$$

The enclosure of the maximizer contains the best packing given by Graham and Lubachevsky (1996).

# References


[1] (2001) Casado, L.G., I. García, P.G. Szabó, and T. Csendes,: Equal Circles Packing in a Square II. - New Results for up to 100 Circles Using the TAMSASS-PECS Algorithm. In: F. Giannessi, P. Pardalos, and T. Rapcsák (eds.): Optimization Theory - Recent Developments from Matrahaza. Kluwer, Dordrecht, 207-224.

[2] R. L. Graham and B. D. Lubachevsky (1996), *Repeated patterns of dense packings of equal disks in a square* Electronic Journal of Combinatorics 3:R16.

[3] C. de Groot M. Monagan, R. Peikert, and D. Würtz, (1992), *Packing circles in a square: review and new results*, in: System Modeling and Optimization, Lecture Notes in Control and Information Services, Vol. 180, 45-54.

[4] R. Hammer, M. Hocks, U. Kulisch and D. Ratz (1993), *Numerical Toolbox for Verified Computing I.*, Springer-Verlag, Berlin.

[5] E. Hansen (1992), *Global Optimization Using Interval Analysis*, Marcel Dekker, New York.

[6] O. Knüppel (1993), *PROFIL – Programmer's Runtime Optimized Fast Interval Library*, Bericht 93.4. Technische Universität Hamburg-Harburg.

[7] M.Cs. Markót, T. Csendes, and A.E. Csallner (1999), *Multisection in Interval Methods for Global Optimization II. Numerical Tests*, Journal of Global Optimization (16), 219-228.

[8] M.Cs. Markót (2000), *An Interval Method to Validate Optimal Solutions of the "Packing Circles in a Unit Square" Problems*, CEJOR (8), 63-78.

[9] Moore R.E. (1966), *Interval Analysis.* Prentice-Hall, Englewood Cliffs, N.J.

[10] K.J. Nurmela and P.R.J. Östergård (1997), *Packing up to 50 circles in a square*, Discrete and Computational Geometry (18), 111-120.

[11] K.J. Nurmela and P. R. J. Östergård (1999) *Optimal packings of equal circles in a square*, in Y. Alavi, D. R. Lick, and A. Schwenk (eds.), Combinatorics, Graph Theory, and Algorithms (Proceedings of the Eighth Quadrennial International Conference on Graph Theory, Combinatorics, Algorithms, and Applications), 671-680.

[12] K.J. Nurmela and P.R.J. Östergård (1999), *More Optimal Packings of Equal Circles in a Square*, Discrete & Computational Geometry (22), 439-457.

[13] P. G. Szabó (2000) *Some New Structures for the "Equal Circles Packing in a Square" Problem*, CEJOR (8), 79-91.