Mihály Csaba Markót

Reliable Global Optimization Methods for Constrained Problems and Their Application for Solving Circle Packing Problems

Abstract of the dissertation

Supervisor:
Dr. Tibor Csendes

University of Szeged, 2003.

1 Introduction

In order to provide easier identification, the numbering of the chapters, definitions, and theorems of this abstract matches their numbering in the dissertation.

Notations

Real numbers and real intervals are denoted by small and capital letters, respectively. Real and interval vectors are denoted by boldface symbols, the components of vectors and the elements of vector and interval sequences are denoted by subscripts. The symbols $\mathbb{R}$ and $\mathbb{I}$ mean the set of real scalars and real intervals, respectively. For a set $D \subseteq \mathbb{R}^n$ the set of n-dimensional boxes $\boldsymbol{X}, \boldsymbol{X} \subseteq D$ is denoted by $\mathbb{I}(D)$. Real-type functions are always identified by small type faces, while their interval inclusion functions are denoted by the corresponding capital letters.

Interval analysis, inclusion functions

Real interval: a non-empty, closed, bounded subset of $\mathbb{R}$, for which $X = [\underline{X}, \overline{X}] = \{x \in \mathbb{R} \mid \underline{X} \leq x \leq \overline{X}\ \}$, where $\underline{X}$ and $\overline{X}$ is the *lower* and *upper* bound of X, respectively. The *width* of an interval is given by $w(X) := \overline{X} - \underline{X}$, the *relative width* of an interval is given by $w_{rel}(X) := w(X)/(\min\{|x| \mid x \in X\})$ if $0 \notin X$, and $w_{rel}(X) := w(X)$ otherwise. The *real-type elementary arithmetic operations* are extended to interval arguments by $X \circ Y := \{x \circ y \mid x \in X,\ y \in Y\} \in \mathbb{I},\ \ \circ \in \{+, -, \cdot, /\}$. The elementary operations on interval vectors are defined componentwise. If $\varphi : D \subseteq \mathbb{R} \to \mathbb{R}$ is a *real, standard mathematical function* which is continuous on each closed interval in D, then the interval extension of φ is defined by $\Phi(X) := \{\varphi(x) \mid x \in X\}$ for each $X \in \mathbb{I}(D)$.

Definition 1 *A function $F : \mathbb{I}^n \to \mathbb{I}$ is said to be an* inclusion function *of $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$ over $X \in \mathbb{I}(D)$, if $\boldsymbol{y} \in \boldsymbol{Y}$ implies $f(\boldsymbol{y}) \in F(\boldsymbol{Y})$ for each $\boldsymbol{Y} \subseteq \boldsymbol{X}$ box.*

Definition 4 *The inclusion function $F : \mathbb{I}^n \to \mathbb{I}$ is said to be* inclusion isotone *over the box $\boldsymbol{X} \in \mathbb{I}^n$, if for any pair of boxes $\boldsymbol{Y}, \boldsymbol{Z} \subseteq \boldsymbol{X}$, the property $\boldsymbol{Y} \subseteq \boldsymbol{Z}$ implies $F(\boldsymbol{Y}) \subseteq F(\boldsymbol{Z})$.*

Definition 5 *The inclusion function $F : \mathbb{I}^n \to \mathbb{I}$* has the convergence order of $\alpha > 0$ *(or shortly, F is* α-convergent*) over $\boldsymbol{X} \in \mathbb{I}^n$, if there exists a positive constant $c \in \mathbb{R}$ such that for any $\boldsymbol{Y} \subseteq \boldsymbol{X}$, $\boldsymbol{Y} \in \mathbb{I}^n$ box*

$$w(F(\boldsymbol{Y})) - w(f(\boldsymbol{Y})) \leq c(w(\boldsymbol{Y}))^\alpha$$

holds, where $f(\boldsymbol{Y})$ is the range of f over $\boldsymbol{Y}$.

Definition 7 *The inclusion function $F : \mathbb{I}^n \to \mathbb{I}$ has the* zero convergence property *over $\boldsymbol{X} \in \mathbb{I}^n$, if $w(F(\boldsymbol{Z}_i)) \to 0$ holds for all the $\{\boldsymbol{Z}_i\}$, $\boldsymbol{Z}_i \in \mathbb{I}^n$, $i = 1, 2, \ldots$ interval sequences for which $\boldsymbol{Z}_i \subseteq \boldsymbol{X}$ and $w(\boldsymbol{Z}_i) \to 0$.*

The investigated optimization problems

Let f and g_j $(j = 1, \dots, r)$ be continuous, real-type functions over the $\boldsymbol{X}_0$ n-dimensional box. The investigated problem types are

$$\min_{\boldsymbol{x} \in \boldsymbol{X}_0} f(\boldsymbol{x}), \tag{$*$}$$

(*bound constrained global optimization problem*) and

$$\begin{array}{ll} \min & f(\boldsymbol{x}), \\ \text{s.t.} & g_j(\boldsymbol{x}) \le 0, \ j = 1, \dots, r, \\ & \boldsymbol{x} \in \boldsymbol{X}_0, \end{array} \tag{$**$}$$

(*inequality constrained global optimization problem*). In both cases the goal is to determine all the global minimizers and the f^* minimum value.

The main steps of the interval branch and bound algorithm

We update two lists in the algorithm: the Worklist (denoted by $\mathcal{L}_\mathcal{W}$) contains subboxes waiting for further processing, while the subboxes satisfying the stopping criterion (as candidates of enclosures of the global minimizers) are stored in the Resultlist ($\mathcal{L}_\mathcal{S}$).

Subdivision direction selection. The leading box $\boldsymbol{Y}$ is subdivided perpendicular to its kth component, where $k := \min \{ j \mid j \in \{1, 2, \dots, n\},\ D(j) = \max_{i=1}^{n} D(i) \}$.

Rule 'A': $D(i) := w(Y_i)$.

Rule 'B': $D(i) := w(\boldsymbol{\nabla}_i F(\boldsymbol{Y})) \cdot w(Y_i)$.

Rule 'C': $D(i) := w(\boldsymbol{\nabla}_i F(\boldsymbol{Y}) \cdot (Y_i - m(Y_i)))$.

Rule 'D': $D(i) := w_{rel}(Y_i)$. ($w_{rel}(X)$ is the relative width of X.)

Subdivision methods. Traditional *bisection* rules: the leading box is divided into two subboxes. *Multisection*: splittings are made in more than one direction at the same time. In Chapter 3 of the dissertation I am dealing with new type of *adaptive* subdivision rules.

Accelerating devices. In the presented algorithms the following accelerating tools are applied: feasibility test, midpoint (cutoff) test, monotonicity test, concavity test, interval Newton-step. Chapter 4 shows some examples of designing problem specific accelerating tests.

Interval selection rules. *Moore–Skelboe rule*: let the leading box be that of the element $\boldsymbol{Y}$ of the Worklist which has the smallest $\underline{F}(\boldsymbol{Y})$ value. *Hansen rule*: let the leading box be the oldest box stored in the Worklist. New type of interval selection rules based on heuristic quantities are studied in Chapter 3.

DEFINITION 9 *Consider the limits of the convergent (nested) subsequences of the sequence of leading boxes $\{\boldsymbol{Y}_i\}_{i=1}^{\infty}$. Let us denote the union of these limits by $A \subseteq \boldsymbol{X}_0$. Then we say that the investigated algorithm variant converges to A.*

2 Multisection subdivision rules

In Chapter 2 of the dissertation several multisection interval subdivision methods were introduced and studied in an empirical way [5]. The so-called multisplitting rules (which are very similar to that of the multisection) and the convergence properties of the algorithm variants using multisplitting were investigated by Csallner et al. [1] in a theoretical way. In the numerical tests I combined the subdivision direction selection rules 'A', 'B', 'C' and 'D' with all of the following subdivision methods:

– /2 rule: the classical bisection;

– /3 rule: subdivision of the leading box into 3 subboxes in the following way: at first, a bisection is performed to the leading box, and secondly, one of the two resulting subboxes is divided by a bisection in the second most promising direction;

– /4 rule: subdivision of the leading box into 4 subboxes by two concurrent bisections in the first and second most promising directions.

Numerical results. I carried out the numerical tests on a large, widely studied standard test set consisting of 37 problems. Summarizing the numerical experiences we can conclude that multisection is advantageous and may result in significant improvements when solving hard to solve problems. According to the majority of indicators, the algorithm variants C/3 and C/4 are the most efficient. As a part of the study, the results of earlier numerical tests showing the advantages of the newer direction selection rules 'B' and 'C' were confirmed, and moreover, it was shown that multisection may additionally improve these advantages. These results are even more remarkable when we take into account that the studied algorithms contained sophisticated accelerating tests. For hard problems, the expected relative improvements of the best multisection algorithm variants are about 22% in the running time and about 25% in the number of objective function evaluations – compared to the best bisection methods. On the other hand, multisection usually resulted in only slightly larger memory complexity than bisection.

3 Study of a new heuristic quantity – a feasibility degree index

In Chapter 3 I introduce a new heuristic quantity for inequality constrained problems [6]. The quantity approximates the feasibility degree of a box, thus, it is suitable to utilize the constraint information both in interval selection and in subdivision rules.

The RejectIndex parameter for bound constrained problems

For $(*)$ problems Casado, Csendes, García and Martínez have recently investigated the heuristic index

$$pf^*(\boldsymbol{X}) = p(f^*, \boldsymbol{X}) = \frac{f^* - \underline{F}(\boldsymbol{X})}{w(F(\boldsymbol{X}))},$$

an indicator of the proximity of a box to a minimizer point. In practice, the global minimum value, f^* is usually not known in advance, thus, an $\hat{f}$ approximation of it must be used. Csendes considered the $\hat{f}$ values as the actual element of a general sequence $\{f_k\}_{k=1}^{\infty}$.

The feasibility degree index

The new parameter is proposed by J. Fernández Hernández: evaluate

$$pu_{G_j}(\boldsymbol{X}) = \min\left\{\frac{-\underline{G_j}(\boldsymbol{X})}{w(G_j(\boldsymbol{X}))}, 1\right\}, \tag{1}$$

for each constraint. When $w(G_j(\boldsymbol{X})) = 0$, g_j is constant over $\boldsymbol{X}$. In this case, if the range $g_j(\boldsymbol{X})$ is positive, then $\boldsymbol{X}$ is eliminated by the feasibility test, otherwise let $pu_{G_j}(\boldsymbol{X}) = 1$. Moreover, let

$$pu(\boldsymbol{X}) = \prod_{j=1}^{r} pu_{G_j}(\boldsymbol{X}), \quad \text{and then} \quad pup(\hat{f}, \boldsymbol{X}) = pu(\boldsymbol{X}) \cdot p(\hat{f}, \boldsymbol{X}),$$

where $\hat{f}$ is determined by a general sequence $pup(\hat{f}, \boldsymbol{X})$. The proposed $pu(\boldsymbol{X})$ parameter justifies the value of the parameter $p(\hat{f}, \boldsymbol{X})$ by a factor measuring the feasibility degree of the box.

The investigated interval selection rules

Classical methods:

– **C1:** Moore–Skelboe rule.

– **C2:** Hansen-rule.

Rules based on $p(f_k, \boldsymbol{X})$:

– **C3:** Select the box from the Worklist which has maximal $p(f_k, \boldsymbol{X})$ value.

– **C4:** A hybrid selection rule using $p(f_k, \boldsymbol{X})$: let N_m be a given positive constant. In each iteration, the algorithm chooses N_m boxes stored in the Worklist, the ones with the largest $p(f_k, \boldsymbol{X})$ values. The updated Worklist consists of the selected boxes. The rest of the boxes are stored in a secondary list and are processed (by using C1) at the end of the algorithm.

The convergence properties of rule C3 were investigated by Csendes, while the computational behavior of rules C3 and C4 was studied by Casado et al.

New rules based on $pup(f_k, \boldsymbol{X})$:

– **C5:** Select the box from the Worklist which has maximal $pup(f_k, \boldsymbol{X})$ value.

– **C6:** A hybrid rule like C4, but using the maximal $pup(f_k, \boldsymbol{X})$.

To select a box which has both a high $pup(f_k, \boldsymbol{X})$ value and a low $\underline{F}(\boldsymbol{X})$ value: let

$$pupb(f_k, \boldsymbol{X}) = \begin{cases} \underline{F}(\boldsymbol{X})/pup(f_k, \boldsymbol{X}) & \text{if } pup(f_k, \boldsymbol{X}) \neq 0, \\ M & \text{otherwise,} \end{cases}$$

where M is a preset large positive value.

– **C7:** Select the box from the Worklist which has *minimal* $pupb(f_k, \boldsymbol{X})$.

– **C8:** A hybrid rule like C4 but using the minimal $pupb(f_k, \boldsymbol{X})$.

The investigated interval subdivision rules

Static bisection and multisection rules:

– **S1_/2:** classical static bisection in the widest component;

– **S2_/4:** static tetrasection, i.e. bisection in the two widest components (/4 multisection);

– **S3_/9:** static multisection into 9 subboxes (trisection in the two widest components).

Adaptive multisection based on $p(f_k, \boldsymbol{X})$**:** Let $0 \le P_1 < P_2 \le 1$ be given threshold values.

– **S4_pf_/2,4,9:**
(a) If $p(f_k, \boldsymbol{X}) < P_1$, then perform a bisection on $\boldsymbol{X}$;
(b) if $P_1 \le p(f_k, \boldsymbol{X}) \le P_2$, then perform a /4 multisection on $\boldsymbol{X}$;
(c) if $p(f_k, \boldsymbol{X}) > P_2$, then perform a /9 multisection on $\boldsymbol{X}$.

Adaptive multisection based on $pup(f_k, \boldsymbol{X})$**:**

– **S5_pup_/2,4,9:** like S4, but using $pup(f_k, \boldsymbol{X})$ instead of $p(f_k, \boldsymbol{X})$.

Rule S4 was recently investigated by Casado et al.

Convergence investigation

Convergence results for the new interval selection rules. In Section 3.3.1 of the dissertation I introduce the convergence properties of the algorithm variants using rules C5 and C7 [6]. In order to investigate the convergence properties, we assume that the stopping condition of the algorithms cannot be fulfilled. I study the algorithms with the feasibility test as the only accelerating device, with the exception of those cases where the applied accelerating tests are explicitly indicated.

THEOREM 17 *[6] Assume that the inclusion function of the objective function is isotone and it has the zero convergence property, the inclusion functions of the constraint functions are isotone, and the* $p(f_k, \boldsymbol{Z})$ *parameters are calculated by the sequence* $\{f_k\}_{k=1}^{\infty}, f_k \to \hat{f} > f^*$*, for which there exists a feasible point* $\hat{\boldsymbol{x}} \in \boldsymbol{X}_0$ *with* $f(\hat{\boldsymbol{x}}) = \hat{f}$*. Then the interval branch-and-bound algorithm that selects that interval from the working list which has the maximal* $pup(f_k, \boldsymbol{Z})$ *value (i.e., which uses criterion C5) may converge to a feasible point* $\hat{\boldsymbol{x}} \in \boldsymbol{X}_0$ *for which* $f(\hat{\boldsymbol{x}}) > f^*$*, i.e., to a point which is not a global minimizer point of the problem.*

In Theorem 18 I prove a similar statement in the case when the inclusion functions of the constraint functions are α-convergent (but not necessarily isotone).

The cutoff test and the zero convergence property of the inclusion function of f *can be* sufficient to ensure convergence to global minimizer points even if f_k does not converge to f^*.

LEMMA 1 *[6] Assume that the interval inclusion functions of the constraint functions are inclusion isotone. Let $\{\boldsymbol{X}_k\}_{k=1}^{\infty}$ be a nested subbox sequence generated by the branch-and-bound algorithm that converges to a feasible point $\boldsymbol{x}$. Then the following three statements are equivalent:*

1. *For a suitable number i, $pu(\boldsymbol{X}_i) = 0$.*

2. *For a suitable number i, $pu(\boldsymbol{X}_k) = 0\ \forall k \geq i$.*

3. *For a suitable i and $\forall k \geq i$ the subbox $\boldsymbol{X}_k$ does not contain strictly feasible points, and moreover, there exists a constraint g_s having $g_s(\boldsymbol{x}) = 0$ and having the G_s inclusion function providing interval inclusions without overestimation on the lower bound over all these $\boldsymbol{X}_k$ subboxes.*

THEOREM 19 *[6] Assume that the inclusion functions of the objective function and the constraint functions have the zero convergence property, no interval sequence $\{\boldsymbol{X}_k\}_{k=1}^{\infty}$, $\boldsymbol{X}_k \subseteq \boldsymbol{X}_0$ contains an $\{\boldsymbol{X}_{k_l}\}$ subsequence for which $\lim_{l\to\infty} pu(\boldsymbol{X}_{k_l}) = 0$ and that f_k converges to $\hat{f} < f^*$. Then the interval branch-and-bound algorithm that selects that box $\boldsymbol{Z}$ from the working list which has the maximal $pup(f_k, \boldsymbol{Z})$ value as first criterion, and the box which has the maximal $p(f_k, \boldsymbol{Z})$ value as second criterion (i.e., for choosing between boxes having the same $pup(f_k, \boldsymbol{Z})$ value) produces an everywhere dense sequence of subintervals converging to each feasible point of the search region $\boldsymbol{X}_0$ regardless of the objective function value.*

THEOREM 20 *[6] Assume that the inclusion functions of both the objective function and the constraint functions are isotone and have the zero convergence property. Consider the interval branch-and-bound algorithm that uses the feasibility test (and optionally the cutoff and the monotonicity tests) and that selects as next actual box that subinterval $\boldsymbol{Z}$ from the working list which has the maximal $pup(f_k, \boldsymbol{Z})$ value as first criterion, and the box which has the maximal $p(f_k, \boldsymbol{Z})$ value as second criterion. If the sequence $\{f_k\}$ converges to the global minimum value f^* and there exist at most a finite number of f_k values below f^*, then the algorithm converges to a set of global minimizer points.*

Applying the above theorem for the updated best upper bound of the global minimum (i.e. for $\tilde{f}$), we can state the following corollary:

COROLLARY 1 *[6] If our algorithm uses the feasibility test (and optionally the cutoff test and the monotonicity test) as accelerated devices, and selects as next leading box that box $\boldsymbol{Z}$ from the working list which has the maximal $pup(\tilde{f}, \boldsymbol{Z})$ value as first criterion, and the box which has the maximal $p(\tilde{f}, \boldsymbol{Z})$ value as second criterion, where $\tilde{f}$ is the best upper bound for the global minimum value, and* its convergence to f^* can be ensured, *then the algorithm converges exclusively to global minimizer points.*

The interval selection rules can be combined through a $u : \mathbb{R}^2 \to \mathbb{R}$ utility function:

THEOREM 21 *[6] Assume that the inclusion functions of both the objective function and the constraint functions are isotone and have the zero convergence property. Consider an interval branch-and-bound algorithm that uses the feasibility test (and optionally the cutoff test and the monotonicity test), and which selects as next actual box that box from the working list which has the maximal $u(pup(f_k, \boldsymbol{Z}), -\underline{F}(\boldsymbol{Z}))$ value as first criterion, and the box which has the maximal $p(f_k, \boldsymbol{Z})$ value as second criterion.*

(a) *Then the sufficient conditions for the convergence of the algorithm to a set of global minimizer points are that the sequence $\{f_k\}$ converges to the global minimum value f^*, there exist at most a finite number of f_k values below f^* and that the utility function $u(x, y)$ is strictly monotonically increasing in both of its arguments.*

(b) *On the other hand, $f_k > f^* + \delta$ for a $\delta > 0$ allows convergence to a non-optimal point even for the class of the above defined utility functions having the property under (a).*

If we additionally assume that we apply the cutoff test with the $\tilde{f}$ values, we get a general convergence condition:

THEOREM 22 *[6] Assume that the inclusion functions of both the objective function and the constraint functions have the zero convergence property. Consider an interval branch-and-bound algorithm that uses the cut-off test, in addition to the feasibility test (and optionally, the monotonicity test) and that uses an arbitrary interval selection criterion (e.g. one of the criteria C1 to C8). Then a sufficient condition for the convergence of this algorithm to a set of global minimizer points is that the sequence $\{f_k\}$ converges to the global minimum value f^* and with the exception of a finite number of iteration steps at each iteration $f_k = f(\boldsymbol{x}_k)$ holds for an $\boldsymbol{x}_k \in \boldsymbol{X}_0$ feasible point.*

If it is ensured, that the sequence of the $\tilde{f}$ values converges to f^*, then, under the assumptions of Theorem 22 the algorithm variants using the C1–C8 rules will converge to a set of global minimizer points.

As we have seen, the availability of a sequence converging to the global minimum is indispensable. One of the common estimators of the global minimum is given by

$$\tilde{f}_k := \min\{\tilde{f}_{k-1}, \overline{F}(\boldsymbol{Y}^1), \ldots, \overline{F}(\boldsymbol{Y}^s)\}, \tag{2}$$

where $\boldsymbol{Y}^1, \ldots, \boldsymbol{Y}^s$ are the result boxes after subdividing the leading box in the iteration cycle k, and $\tilde{f}_0 := \overline{F}(\boldsymbol{X}_0)$. Another estimator of the global minimum is

$$f_k := \underline{F}(\boldsymbol{X}_k), \tag{3}$$

where $\boldsymbol{X}_k$ is the leading box in the iteration cycle k.

Corollary 2 *[6] Assume that the inclusion functions of both the objective function and the constraint functions are isotone and have the zero convergence property. Consider the interval branch-and-bound algorithm that uses the feasibility test (and optionally the cutoff and the monotonicity tests) and that selects the next box to be subdivided by one of the following rules:*

- *maximal* $pup(f_k, \boldsymbol{X})$ *value (Rule C5),*
- *maximal* $u(pup(f_k, \boldsymbol{X}), -\underline{F}(\boldsymbol{X}))$ *value, or*
- *minimal* $pupb(f_k, \boldsymbol{X})$ *value (Rule C7).*

Then both $\{\tilde{f}_k\}$ *of (2) and* $\{\underline{F}(\boldsymbol{X}_k)\}$ *of (3) may fail to converge to* f^*.

Properties of the subdivision rules. In Section 3.3.2 of the dissertation I study the adaptive multisection rule based on the parameter $pup(\tilde{f}, \boldsymbol{X})$ [6]. The $\tilde{f}$ values are assumed to be generated by (2).

THEOREM 23 *[6] Let us consider an interval B&B algorithm which selects the box with the smallest lower bound* $\underline{F}(\boldsymbol{X})$ *as the next box to be subdivided (i.e. applies Rule C1). There exist optimization problems (**) for which the inclusion functions of both the objective function and the constraint functions are isotone and* α*-convergent, and the following statements are true:*

1. *An arbitrary large number* $N(> 0)$ *of consecutive leading boxes selected by the algorithm have the properties that: neither of these processed boxes* $\boldsymbol{X}$ *contains a global minimizer point, and the related* $pup(\tilde{f}, \boldsymbol{X})$ *values are larger than a preset* $P_2 < 1$.
2. *There exists a subsequence of the leading intervals converging to a global minimizer point, such that for each element of this subsequence* $pup(\tilde{f}, \boldsymbol{X}) < P_1$ *holds for a fixed* $0 < P_1$.

Computational studies

In Section 3.4 of the dissertation I perform a detailed comparative study between the earlier and the newly proposed algorithm variants. The following test problem are considered:

1. Obnoxious facility location problems.

2. Constrained problems based on a set of standard test functions for bound constrained global optimization.

Studying the interval selection strategies. The generated test problems were solved by the algorithm variants including rules C1–C8. As a brief conclusion we can state that the new type of methods (C5 to C8) may improve the efficiency of the branch and bound algorithm. This improvement can be achieved even in cases when the problem is hard to solve and the rules using $p(f_k, \boldsymbol{Z})$ alone are also very efficient.

Moreover, in many cases the hardest problems could be solved only with those algorithms which utilized the constraint information during the interval selection step. The improvement is especially considerable for problems in which the optimizers are situated close to the border of the feasible set

Studying the adaptive multisection rules. The subdivision rules S1–S5 were all examined for both the facility location problems and the constrained versions of the standard test problems. As the results show, the adaptive S4 rule proved to be slightly better than rule S5. The reason of these – mainly negative – results is probably the presetting methodology of the P_1 and P_2 threshold values: for the more sophisticated S5 subdivision methods the threshold parameters should be determined with more accurate methods.

4 Solving circle packing problems

In Chapter I am dealing with the problem of finding the optimal packings of congruent circles in a square. In contrast to the recent computer methods, the present algorithms (used as computer-assisted optimality proofs) are fully based on interval computations. In the bulk of the chapter I discuss the construction of some elements of the algorithmic frame. These tools are designed specifically for the considered problem class.

Problem definitions

Let $n \geq 2$ be a given integer. The commonly used approaches of the particular packing problem are the following:

1. *Place a number of n equal circles without overlapping into the unit square maximizing the radius of the circles.*

The following approach results in an equivalent optimization problem:

2. *Place a number of n points into the unit square maximizing the minimal distance between the pairs of points*

In the dissertation I study the 2. representation of the problem considering square of distances instead of distances. Let the unit square be the set $[0,1]^2$, and denote $((x_1, y_1), \ldots, (x_n, y_n))$ the set of points to be locate. (In the sequel I denote this set briefly by $(\boldsymbol{x}, \boldsymbol{y}) \in [0,1]^{2n}$). Moreover, let d_{ij} denote the square of distance between the points (x_i, y_i) and (x_j, y_j). Then the global optimization problem is the following:

$$\begin{array}{ll} \max & \min_{1 \leq i \neq j \leq n} d_{ij}, \\ \text{s.t.} & 0 \leq x_i, y_i \leq 1, \quad i = 1, 2, \ldots, n. \end{array} \tag{4}$$

The objective function of (4) to be maximized is as below:

$$f_n(\boldsymbol{x}, \boldsymbol{y}) = \min_{1 \leq i \neq j \leq n} (x_i - x_j)^2 + (y_i - y_j)^2 = \min_{1 \leq i \neq j \leq n} d_{ij}. \tag{5}$$

Prior to the results of the present dissertation the optimal solutions of the cases $n = 2, \ldots, 27, 36$ were known. A part of the optimality proofs based on computer-assisted methods, but those methods used floating point arithmetic.

Introducing an interval inclusion function

In Section 4.3 I give a non-trivial interval inclusion function of the objective function (5):

THEOREM 24 *[2] Let $(\boldsymbol{X}, \boldsymbol{Y}) \subseteq [0,1]^{2n}$, and let*

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

Let $a := \min_{1 \le i \ne j \le n} \underline{D_{ij}}$, $a \in \mathbb{R}$, and $b := \min_{1 \le i \ne j \le n} \overline{D_{ij}}$, $b \in \mathbb{R}$. Then the interval $F_n(\boldsymbol{X}, \boldsymbol{Y}) := [\, a, b \,]$ encloses the range of $f_n(\boldsymbol{x}, \boldsymbol{y})$ over the $(\boldsymbol{X}, \boldsymbol{Y})$ $2n$-dimensional box.

I prove the following important properties of the constructed inclusion function:

– An evaluation of $F_n(\boldsymbol{X}, \boldsymbol{Y})$ requires $\Theta(n^2)$ operations (similarly to the evaluation of $f_n(\boldsymbol{x}, \boldsymbol{y})$).

– for $n = 2$ $F_n(\boldsymbol{X}, \boldsymbol{Y})$ provides sharp enclosure in both bounds (apart from the directed outward roundings of the interval arithmetic implementation); but in the case of $n \ge 3$ $F_n(\boldsymbol{X}, \boldsymbol{Y})$ is sharp only in the lower bound, in the upper bound it may result in overestimation;

– $F_n(\boldsymbol{X}, \boldsymbol{Y})$ is inclusion isotone;

– $F_n(\boldsymbol{X}, \boldsymbol{Y})$ is zero convergent.

The first version of the developed algorithm

In Section 4.4 I give an overview on the further specification of the frame algorithm:

Monotonicity properties.

THEOREM 25 *[2] Let $(\boldsymbol{X}, \boldsymbol{Y}) \subseteq [0,1]^{2n}$ be an arbitrary $2n$-dimensional box. If for some k, $k \in \{1, 2, \ldots, n\}$ and for all $j (\ne k)$ either $\overline{X_k} \le \underline{X_j}$, or $\underline{D_{kj}} > \overline{F_n}(\boldsymbol{X}, \boldsymbol{Y})$ holds, then f_n is monotonously decreasing over $(\boldsymbol{X}, \boldsymbol{Y})$ in the variable x_k. Then X_k can be shrinked to $[\underline{X_k}, \underline{X_k}]$.*

The objective function can be tested in a similar way for the monotonous increasing property and for the variables y_k.

The method of handling free circles.

DEFINITION 12 *Consider the point packing problem (4) for a given $n \ge 2$. Consider an optimal point packing, i.e. a vector $(\boldsymbol{x}, \boldsymbol{y}) = (x_1, \ldots, x_n, y_1, \ldots, y_n)$ for which f_n is maximal. We call a point $p_k = (x_k, y_k)$, $k \in \{1, \ldots, n\}$ of this optimal packing to* *be a* free point, *if there exists a half line H with the endpoint p_k and there exists a positive real number ε, such that*

$$f_n(\boldsymbol{x}, \boldsymbol{y}) = f_n(x_1, \ldots, x'_k, \ldots, x_n, y_1, \ldots, y'_k, \ldots, y_n)$$

for all $(x'_k, y'_k) \in H \cap \Delta_\varepsilon(p_k)$, where $\Delta_\varepsilon(p_k)$ denotes the ε-neighborhood of p_k.

Each free point in an optimal point packing is identified by the center of a *free circle* of the corresponding optimal circle packing. The handling of free circles is crucial in the solution of circle packing problems, since they pose a continuum set of equivalent global optimizers.

THEOREM 27 *(based on) [2] Let $(\boldsymbol{X}, \boldsymbol{Y}) \subseteq [0,1]^{2n}$ be an arbitrary $2n$-dimensional box. If for some k, $k \in \{1, 2, \ldots, n\}$ and for all $j (\neq k)$ the statement $\underline{D_{kj}} > \overline{F_n}(\boldsymbol{X}, \boldsymbol{Y})$ holds, then for each optimal packing in $(\boldsymbol{X}, \boldsymbol{Y})$ the point (x_k, y_k) of the packing is a free point and it can move within (X_k, Y_k). Thus, for further investigations on the box it is enough to store one point from (X_k, Y_k) together with the information '(X_k, Y_k) can contain exclusively free points'.*

The method of active areas. This method played a key role both in the earlier theoretical and in the computer-aided optimality proofs. The developed reliable variants of the method proved to be the most efficient accelerating tests of the present algorithms. The essence of the method is the following: Assume that we have a $\tilde{f}_0$ lower bound for the maximum of the minimal pairwise distances. Then, from each component (X_i, Y_i) of the actual box we can iteratively delete those points, which have a distance smaller than $\tilde{f}_0$ to *all* points of the remained region of another component.

As a starting point, I represented the remaining region of each component by a rectangle in the intermediate steps. Under this condition I developed a basic interval elimination process between two rectangles. In order to improve the efficiency of the method, in the first algorithm I quantized each rectangle horizontally and vertically into cells, and I performed the basic elimination procedure for the relevant *rows of cells* and *columns of cells*, respectively.

Numerical results for local verification. In the first numerical tests my goal was to carry out an interval-based reliable validation of the earlier published real-type optimal solutions for $n = 2, \ldots, 27, 36$. Note that a part of these solutions were achieved by computer methods using floating point operations. The width of the components of the starting box was set to 0.01. The possible results of the verification process are the following:

– *Acceptation*: the enclosure of the maximum value contains the published real-type maximum, and moreover, an element of $\mathcal{L}_\mathcal{S}$ encloses the given real-type solution.

– *Rejection*: the algorithm terminates with the empty lists $\mathcal{L}_\mathcal{W}$ and $\mathcal{L}_\mathcal{S}$. Then we can conclude that in the initial search region there exists no solution having the pre-given objective function value.

The introduced algorithm was able to verify (accept) – with the exception of 4 hard cases – the optimality of all the previously known real-type optimizers and optimum values within 2 hours of running time. As a try, the test set contained a false optimum value from an older publication. This packing value was correctly rejected by the algorithm.

Numerical results for global verification. The hardest difficulty was the efficient handling of such solutions which are equivalent in the geometric sense but which differ in their numerical representation (e.g. results of index permutations or symmetry transformations). The currently known best idea resolving this difficulty is the splitting of the square into tiles. All the optimal solutions can be found by processing all the tile combinations consisting of n tiles. Both in the earlier proposed and the present algorithms the square is divided into $k \times l$ identical rectangles.

During the global verification procedures the time limit was set to 4 hours. The pre-given optima and optimizers to be checked were the same as for the local verification study. With the exception of 3 cases, for $n = 2, \ldots, 20$ the algorithm has managed to validate (accept) the correctness of the real-type solutions (within the applied tolerance value). In each unsolved case the main difficulty was to carry out the procedure on one particular tile combination.

The improved algorithm

As a result of my more extensive study [3, 4] I developed an improved algorithm of the basic algorithm of Section 4.4 which is able to solve some open problem instances $n \geq 28,\ n \neq 36$. This algorithm is introduced in Section 4.5.

A more sophisticated method for handling free circles.

1. Let $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathbb{I}^{2n}$ include all the remaining boxes after a certain number of iteration loops when executing the B&B algorithm. Let $\tilde{f}$ be the current cutoff value.
2. Assume that there exist *machine representable points* $p_{k_1}, \ldots, p_{k_t},\ p_{k_s} \in (X_{k_s}, Y_{k_s})$, $s \in \{1, \ldots, t\}$ within t different components of $(\boldsymbol{X}, \boldsymbol{Y})$ such that $\underline{D}(p_{k_s}, (X_j, Y_j)) > \overline{F}(\boldsymbol{X}, \boldsymbol{Y}) \geq \tilde{f}$ holds for all $s \in \{1, \ldots, t\}$ and for all $j \neq k_s,\ j \in \{1, \ldots, n\}$. Let K denote the set of indices $\{k_1, \ldots k_t\}$.
3. Replace the components (X_i, Y_i) with the point intervals p_i for each $i \in K$. Run the B&B algorithm on the resulting $(\boldsymbol{X}', \boldsymbol{Y}')$ box ignoring the step of improving $\tilde{f}$.
4. Let $(\boldsymbol{X}'', \boldsymbol{Y}'') \in \mathbb{I}^{2n}$ include all the candidate boxes remained. The output box of the procedure is then given by (X_i, Y_i) for $i \in K$ and (X''_j, Y''_j) for $j \notin K$.

THEOREM 28 *[4] The above procedure is correct in the sense that all the optimal solutions in $(\boldsymbol{X}, \boldsymbol{Y})$ are also present in the output box.*

Improving the method of active areas: eliminating with polygon representations. Instead of representing the remaining regions by sets of cells, Nurmela and Östergård approximated the remaining sets by polygons in their computer-assisted method. The proposed procedure raises several problems when using floating point computations. As a solution I introduce the main features of a reliable version of this polygon approach using interval computations.

Investigating subsets of tile combinations. Prior to the results of the present dissertation, the main problem when solving circle packing problem instances for

$n > 27$, $n \neq 36$ was the number of initial tile combinations. (For $n = 28$, a sequential process on those combinations would require about 1000 times more processor time – i.e. about several decades, even with non-interval computations – compared to the case of $n = 27$.) The idea behind the newly proposed method is that we can utilize the local relations between the tiles and eliminate groups of tile combinations together. Let us denote a generalized point packing problem instance by $P(n, X_1, \ldots, X_n, Y_1, \ldots, Y_n)$ where n is the number of points to be located, $(X_i, Y_i) \in \mathbb{I}^2$, $i = 1, \ldots n$ are the components of the starting box, and the objective function of the problem is given by (5). The theorem below shows how to apply a result achieved on a $2m$-dimensional packing problem for a $2n$-dimensional problem with $n \geq m \geq 2$.

THEOREM 30 *[4] Let $n \geq m \geq 2$ be integers and let*

$$P_m = P(m, Z_1, \ldots, Z_m, W_1, \ldots, W_m) = P(m, (\boldsymbol{Z}, \boldsymbol{W})), \text{ and}$$
$$P_n = P(n, X_1, \ldots, X_n, Y_1, \ldots, Y_n) = P(n, (\boldsymbol{X}, \boldsymbol{Y}))$$

be point packing problem instances ($X_i, Y_i, Z_i, W_i \in \mathbb{I}; X_i, Y_i, Z_i, W_i \subseteq [0, 1]$). Run the B&B algorithm on P_m using an $\tilde{f}$ cutoff value in the accelerating devices but skipping the step of improving $\tilde{f}$. Stop after an arbitrary number of iteration steps. Let $(Z'_1, \ldots, Z'_m, W'_1, \ldots, W'_m) := (\boldsymbol{Z}', \boldsymbol{W}')$ be the enclosure of all the elements placed on $\mathcal{L}_\mathcal{W}$ and $\mathcal{L}_\mathcal{S}$. Assume that there exists a φ invertable, distance-preserving geometric transformation satisfying $\varphi(Z_i) = X_i$ and $\varphi(W_i) = Y_i$, $\forall i = 1, \ldots, m$. Then for each point packing $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^{2n}$ satisfying $(\boldsymbol{x}, \boldsymbol{y}) \in (\boldsymbol{X}, \boldsymbol{Y})$ and $f_n(\boldsymbol{x}, \boldsymbol{y}) \geq \tilde{f}$, the statement

$$(\boldsymbol{x}, \boldsymbol{y}) \in (\varphi(Z'_1), \ldots, \varphi(Z'_m), \ldots, X_n, \varphi(W'_1), \ldots, \varphi(W'_m), \ldots, Y_n) := (\boldsymbol{X}', \boldsymbol{Y}')$$

also holds.

COROLLARY 3 *[4] Let φ be the identity transformation and assume that the B&B algorithm terminates with an empty Worklist and with an empty Resultlist, i.e. the whole search region $(\boldsymbol{Z}, \boldsymbol{W}) = (Z_1, \ldots, Z_m, W_1, \ldots, W_m) = (X_1, \ldots, X_m, Y_1, \ldots, Y_m)$ is eliminated by the accelerating devices using $\tilde{f}$. Then $(\boldsymbol{X}, \boldsymbol{Y})$ does not contain any $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^{2n}$ vectors for which $f_n(\boldsymbol{x}, \boldsymbol{y}) \geq \tilde{f}$ holds.*

Basic algorithms used in the optimality proofs. I start the optimality proofs by finding the feasible tile patterns and their remaining areas on some small subsets of the whole set of tiles. Then I process bigger and bigger subsets while using the results of the previous steps. Thus, the whole method consists of several phases. The two basic procedures are:

– Grow(): add tiles from a new column to each element of a set of tile combinations.

– Join(): join the elements ot two sets of tile combinations pairwise.

Numerical results: optimal packing of 28, 29, and 30 circles. The obtained results can be summarized as below ($n = 28, 29, 30$):

– Apart from symmetric cases, one initial tile combination (more precisely, the remaining areas of the particular combination) contains all the global optimal solutions of the packing problem of n points.

– The guaranteed enclosures of the global maximum values are

$$
\begin{aligned}
F^*_{28} &= [\underline{0.23053549364266}73, \underline{0.23053549364267}43], \quad w(F^*_{28}) \approx 7 \cdot 10^{-15},\\
F^*_{29} &= [\underline{0.2268829007442}089, \underline{0.2268829007442}240], \quad w(F^*_{29}) \approx 2 \cdot 10^{-14},\\
F^*_{30} &= [\underline{0.22450296453108}81, \underline{0.22450296453109}03], \quad w(F^*_{30}) \approx 2 \cdot 10^{-15}.
\end{aligned}
$$

– The exact global maximum value differs from the currently best known function value by at most $w(F^*_n)$.

– Apart from symmetric cases, all the global optimizers of the problem of packing n points are located in the introduced $(\boldsymbol{X}, \boldsymbol{Y})^*_n$ box. Each component of the result boxes are very narrow (with the exception of the components enclosing possibly free points).

– The differences between the volume of the starting and the result boxes are more than 711, 764, and 872 orders of magnitudes, respectively.

– The total computational time was approximately 53, 50, and 20 hours, respectively.

Optimality of the conjectured best structures. A packing structure specifies which points are located on the sides of the square, which pairs have minimal distance, and which points can move in the particular packing. The vectors corresponding to the rigid subset of a packing are denoted by an r superscript. In the dissertation I prove the following statements in a numerical way:

– The system of equations describing the rigid subset of the particular structure has exactly one $(\boldsymbol{x}, \boldsymbol{y})^r_n$ solution over the box $(\boldsymbol{X}, \boldsymbol{Y})^{*,r}_n$.

– For $n = 28, 29$ the packing $(\boldsymbol{x}, \boldsymbol{y})^r_n$ can be extended by a point which can move in the corresponding component of $(\boldsymbol{X}, \boldsymbol{Y})^*_n$.

– $(\boldsymbol{x}, \boldsymbol{y})^r_n$ is the only optimal point packing in $(\boldsymbol{X}, \boldsymbol{Y})^{*,r}_n$.

References

[1] A.E. Csallner, T. Csendes, and M.Cs. Markót (2000), *Multisection in interval branch–and–bound methods for global optimization. I. Theoretical results*, J. Global Optimization 16, 371–392.

[2] M.Cs. Markót (2000), *An interval method to validate optimal solutions of the "packing circles in a unit square" problems*, Central European Journal of Operations Research 8, 63–78.

[3] M.Cs. Markót, *Optimal Packing of 28 Equal Circles in a Unit Square – the First Reliable Solution.* Accepted for publication in Numerical Algorithms. Available at `http://www.inf.u-szeged.hu/~markot/opt28.ps.gz`.

[4] M.Cs. Markót and T. Csendes, *A New Verified Optimization Technique for the "Packing Circles in a Unit Square" Problems.* Submitted for publication. Available at `http://www.inf.u-szeged.hu/~markot/impcirc.ps.gz`.

[5] M.Cs. Markót, T. Csendes, and A.E. Csallner (2000), *Multisection in interval branch–and–bound methods for global optimization. II. Numerical tests*, Journal of Global Optimization 16, 219–228.

[6] M.Cs. Markót, J. Fernández, L.G. Casado, and T. Csendes, *New interval methods for constrained global optimization*. Conditionally accepted for publication in Mathematical Programming. Available at
`http://www.inf.u-szeged.hu/~markot/mathprog.ps.gz`.