UDC 519.85

Yu. G. Stoyan, Corr. member of NAS of Ukraine
A. M. Chugay, PhD

Institute for Mechanical Engineering Problems
of the National Academy of Sciences of Ukraine
(Kharkov, E-mail: stoyan@ipmach.kharkov.ua; chugay@ipmach.kharkov.ua)

AN OPTIMIZATION PROBLEM OF PACKING IDENTICAL CIRCLES INTO A MULTIPLY CONNECTED REGION

Part 2. A solution method and its realisation

Рассматривается оптимизационная задача упаковки одинаковых кругов в многосвязную область, граница которой состоит из отрезков дуг окружностей и отрезков прямых. На основании свойств математической модели предлагается метод решения задачи. Метод предполагает комбинацию метода получения начальных точек, модифицированного метода возможных направлений для поиска локальных максимумов и модифицированного метода сужающихся окрестностей для поиска приближения к глобальному максимуму. Приводятся численные примеры.

Розглядається оптимізаційна задача пакування однакових кіл у багатозв'язну область, границя якої складається з відрізків дуг околів та відрізків прямих. На підставі властивостей математичної моделі пропонується метод розв'язання задачі. Метод передбачає комбінацію методу одержання початкових точок, модифікованого методу можливих напрямів для пошуку локальних максимумів та модифікованого методу звужувальних околів для пошуку наближення до глобального максимуму. Наводяться числові приклади.

The paper deals with an optimization problem of packing identical circles into a multiply connected region whose frontier consists of arcs of circles and line segments. On the ground of the characteristics of a mathematical model a solution method is offered. The method consists of a combination of a method of generating starting points, a modification of the feasible directions method to search for local maxima and a modification of the decremental neighbourhood search method to find an approximation to a global maximum. Numerical examples are given.

The statement of the considered problem and its mathematical model is presented in [1]. In this paper on the basis of characteristics of the mathematical model a solution method is offered.

In what follows, problems given in [1] we denote as follows:

a) the problem (1)-(2) – the problem *A*;
the problem (3) – the problem *B*.

General solution strategy

According to the mathematical model a solving of the problem is reduced to a performance of some stages. At each of ones the problem *B* of packing *n* circles of the given radius *r* is solved. Firstly, a preliminary estimation of the number of circles to be packed is realized. By virtue of item 9 of mathematical characteristics given in [1] we search for an approximation to a global maximum of the problem *B* using the decremental neighbourhood search method [2, 3] (in what follows DNS). Random vectors which define starting locations of circles into the region are generated by the DNS. On the basis of the vectors starting points from the feasible region are constructed and local maxima of the problem *B* are computed. At each stage of the DNS the best local maximum is stored. If process obtaining the best local maximum is decelerated then solution process converges quickly to some local maximum. This local maximum is taken as an approximation to a global one.

Starting from characteristics above for successfully solving the problem A it needs to fulfil the following steps.

Step 1. Define n_0 so that a containment of n_0 circles into P is guaranteed. Then take $n = n_0 + 1$ and pass to the next step.

Step 2. Give $r_i = r/2$, and generate $u_i^{nl} \in P$, $i \in I_n$, $l \in I_\lambda = (1,2,...,\lambda)$, by a random way utilizing a modification of the DNS so that $X^{nl} = (u_1^{nl}, u_2^{nl}, ..., u_n^{nl}, \underbrace{\frac{1}{2}r, \frac{1}{2}r, ..., \frac{1}{2}r}_{n}) \in W_n$.

Step 3. Take the points X^{nl} as starting points and search for local maxima X^{nl*}, $l \in J_\lambda$, of the problem B.

Step 4. Select the best local maximum $\widetilde{X}^{n*} = \arg\max\left\{F_n(X^{nl*}),\ l \in J_\lambda\right\}$.

Step 5. If $F_n(\widetilde{X}^{n*}) = nr$ then a global maximum of the problem B is achieved. In this case take $n = n + 1$ and solve the problem B again, i.e. return to the step 2.

Step 6. If $F_n(\widetilde{X}^{n*}) < nr$ and a stopping criterion of the DNS is not fulfilled then return to the step 2.

Step 7. If the stopping criterion of the DNS is fulfilled and $F_n(\widetilde{X}^{n*}) < nr$, then take $u^* = \widetilde{u}^{(n-1)*}$ as an approximation to a global maximum of the problem A and the number of circles packed is $n - 1$.

Construction of starting point

For obtaining of starting points the rectangular lattice is constructed. The length of lattice cell sides is accepted equal to $r/2$. This lattice arbitrarily is placed with respect to the region. Cells which completely belong to the region are selected. The number k of the cells is strictly more than n. Centre coordinates of the cells form a vector p of length $2k$. These vectors are generated by the DNS.

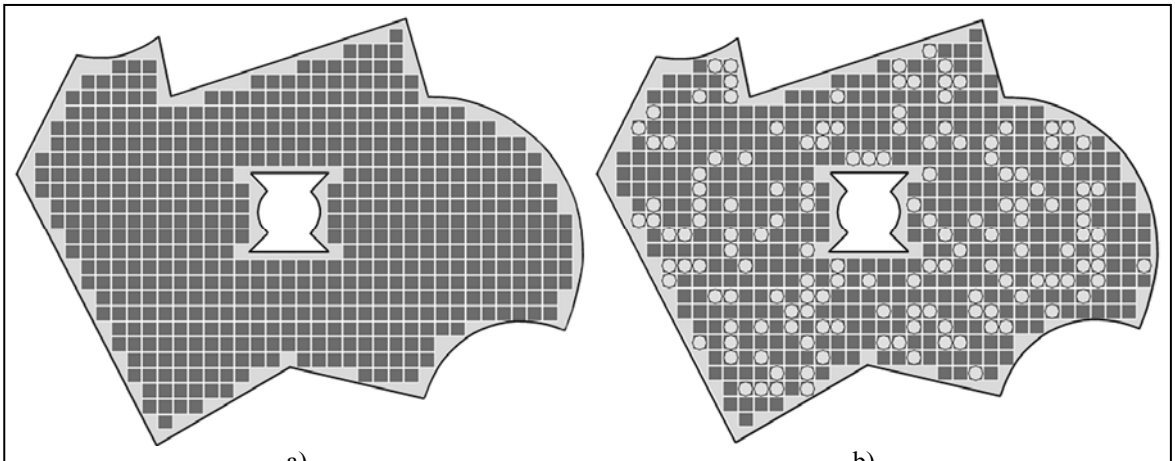
The first $2n$ coordinates of vectors of kind p define the centers of circles and form vector u^n. Then vectors v^n and u^n define starting points $X^n = (u^n, v^n)$.

A starting point is constructed according to the following algorithm (fig.1).

Step 1. Cover the region P by a lattice whose basic parallelogram (cell) is a square

$$S = \left\{(x,y) \in R^2 : -\frac{1}{2}r \le x \le \frac{1}{2}r, -\frac{1}{2}r \le y \le \frac{1}{2}r\right\}.$$

Step 2. Define the number $k > n$ of squares such that



a) b)

Fig. 1. A starting point construction:
a) – lattice construction; b) – placing of circles of radius $r/2$ in lattice cells

$$S_i = \left\{(x,y) \in R^2 : a_i - \frac{1}{2}r \le x \le a_i + \frac{1}{2}r, b_i - \frac{1}{2}r \le y \le b_i + \frac{1}{2}r\right\} \subset P,$$

where (a_i, b_i) are centre coordinates of S_i, $i \in I_k = \{1,2,...,k\}$ (fig. 1, a).

Step 3. Give randomly a vector $p^i = (p_{i_1}, p_{i_2},..., p_{i_n},..., p_{i_k}) \in R^{2k}$, where $p_{i_j} = (a_{i_j}, b_{i_j})$.

It is evident that the number of the vectors p^i is equal $k!$, i. e. the vectors form a permutation set $\Pi \subset R^{2k}$ without repetitions.

Step 4. Give $v^{ni} = (\underbrace{r/2, r/2,..., r/2}_{n})$ and form a vector $u^{ni} = (u_1^{ni}, u_2^{ni},..., u_n^{ni}) \in R^{2n}$ so that $u_j^{ni} = p_{i_j} = (a_{i_j}, b_{i_j})$, $j \in I_n$, i. e. u^{ni} is obtained as a result of equating sequentially components of u^{ni} to the first n components of $p^i = (\underbrace{p_{i_1}, p_{i_2},..., p_{i_n}}_{n},..., p_{i_k})$. Hence, the number of u^{ni} is equal to $\frac{k!}{n!(k-n)!}$.

Thus, taken (a_{i_j}, b_{i_j}) as centre coordinates of C_j we obtain a placement of C_j of radius $r/2$, $j \in I_n$, into P without overlapping (fig. 1, b).

Points of kind $X^{ni} = (u^{ni}, v^{ni}) \in R^{3n}$ are taken as starting points to compute local maxima of the problem B.

To the permutation set $\Pi \subset R^{2k}$ there corresponds a set $\mathrm{T} \subset R^{3n}$ of starting points and, hence, a set $L \subset R^{3n}$ of local maxima of the problem B. Thus, a non-exhaustive search of local maxima of set L can be reduced to a non-exhaustive search of points of $\Pi \subset R^{2k}$.

Local optimization

By virtue of items 6–8 of characteristics of the problem B given in [1] a feasible region can be presented as a finite union of subregions which are described by systems of nonlinear inequalities. It allows to reduce a search of a local maximum of the problem B to a computation of sequence of local maxima on the subregions. To this end we single out one of subregions which contains a starting point. On this subregion searching for a local maximum is carried out. A local maximum obtained can belongs to several subregions. If there exist a subregion for which the point is not a local maximum then the point is taken as a starting point for searching for a new local maximum on the subregion. If such subregion is absent then the point is a local maximum of the problem B.

A search of a local maximum of the problem B can be reduced to solving the following sequence of nonlinear programming problems (NLP)

$$F_n(\overline{X}^{nj*}) = \max F_n(X^n), \text{ s. t. } X^n \in W_{nt_j}, j = 1,2,...,m << \eta_0, \tag{1}$$

$$W_{nt_j} = \{X^n \in R^{3n} : \Gamma_i^{s_t}(u_{i_j}, r_{i_j}) \ge 0,\ i \in I_n, \Phi_{ij}(u_i, u_j, r_i, r_j) \ge 0,\ i, j \in I_n, i < j,\ r - r_i \ge 0, i \in I_n, r_i \ge 0, i \in I_n\}. \tag{2}$$

To solve the NLP (1)–(2) the modification of the Zoutendijk method of feasible directions [4] together with the concept of ε-active inequalities [2, 5] are used. The modification realizes the usual iterative process

$$X^{n(k+1)} = X^{nk} + tZ^k, k = 1,2,...,\upsilon,$$

where $Z^k \in R^{3n}$ is a solution of the following linear programming problem

$$\max \alpha^k, \text{ s.t. } (\alpha^k, Z^k) \in G^k, \tag{3}$$

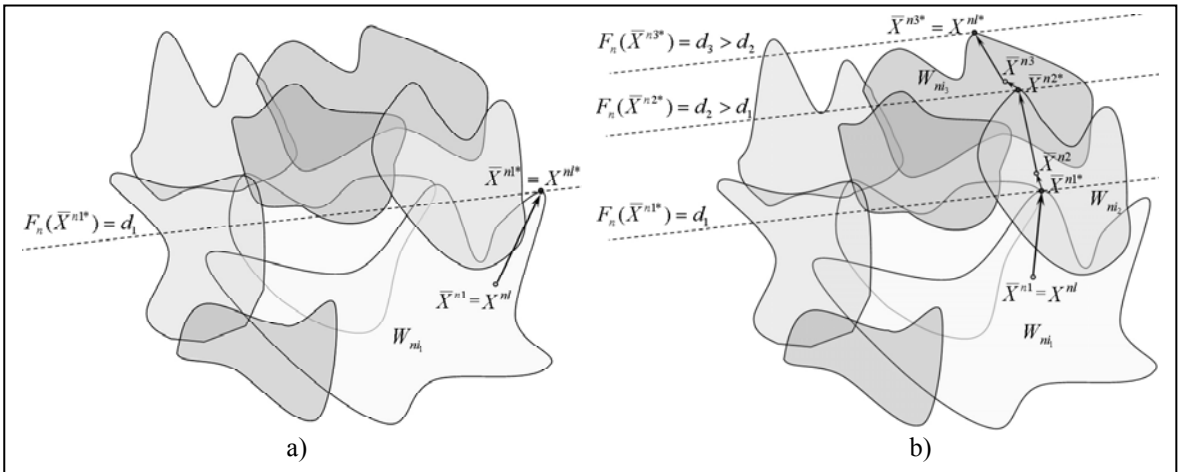


Fig. 2. A scheme of local optimization:

a) – $\bar{X}^{n1*}$ is a local maximum with respect to W_n; b) – $\bar{X}^{n1*}$ is a local maximum with respect to W_{ni_1}

$$G^k = \left\{(\alpha^k, Z^k) \in R^{3n+1} : \left(\nabla F_n(X^{nk}), Z^k\right) \geq \alpha^k, \left(\nabla \Psi_{k_j}(X^{nk}), Z^k\right) \geq w_{k_j}, \right.$$
$$\left. j = 1,2,...,\varsigma_k(\varepsilon_k), -1 \leq z_i^k \leq 1, i = 1,2,...,3n\right\}, \quad (4)$$

$\left(\nabla F_n(X^{nk}), Z^k\right)$ is the scalar product of the gradient of $F_n(X^n)$ and the vector Z^k, $\left(\nabla \Psi_{k_j}(X^{nk}), Z^k\right)$ is the scalar product of the gradient of $\Psi_{k_j}(X^n)$ that is the left side of an ε-active inequality of a system of kind (4) (given in [1]) and Z^k, $w_{k_j} = \alpha^k$ if $\Psi_{k_j}(X^{nk})$ is a concave function and $w_{k_j} = 0$ if otherwise. The problem (3)-(4) is solved by the interior point method [6].

A scheme of local optimization is presented in fig. 2. A transition from one problem of form (1)–(2) to another is carried out as follows. Let $X^{nl} \in W_n$ be a starting point. We single out an inequality system of kind (4) (given in [1]) which specifies a subregion $W_{ni_1} \subset W_n$ such that $X^{nl} \in W_{ni_1}$. Taken X^{nl} as a starting point we solve the problem

$$F_n(\bar{X}^{n1*}) = \max F_n(X^n), \text{ s.t. } X^n \in W_{ni_1}.$$

The point $\bar{X}^{n1*}$ can be a local maximum with respect to either W_n (fig. 2, a) or W_{ni_1} (fig. 2, b).

In order to define whether $\bar{X}^{n1*}$ is a local maximum with respect to W_n (i. e. a local maximum of the problem *B*) we single out ε-active inequalities at the point $\bar{X}^{n1*}$ from the system (2) (given in [1]) and solve the following linear programming problem

$$\max \alpha, \text{ s. t. } (\alpha, Z) \in G,$$
$$G = \left\{(\alpha, Z) \in R^{3n+1} : \left(\nabla F_n(\bar{X}^{n1*}), Z\right) \geq \alpha, \left(\nabla \Psi_j(\bar{X}^{n1*}), Z\right) \geq w_j, \right.$$
$$\left. j = 1,2,...,\varsigma(\varepsilon), -1 \leq z \leq 1, i = 1,2,...,3n\right\}.$$

If $\alpha \leq 0$, then $\bar{X}^{n1*}$ is a local maximum of the problem *B*. If $\alpha > 0$, then $\bar{X}^{n1*}$ is not a local maximum of the problem *B* (fig. 2, b) and it enables to compute a point $\bar{X}^{n1} = (\bar{X}^{n1*} + tZ) \in W_n$ at which $F_n(\bar{X}^{n1*}) < F_n(\bar{X}^{n1})$.

After that we form a new inequality system of kind (4) (given in [1]) that specifies a new subregion $W_{ni_2} \subset W_n$ such that $\bar{X}^{n1} \in W_{ni_2}$. Taken the point $\bar{X}^{n1}$ as a starting point we solve the following NLP:

$$F_n(\bar{X}^{n2*}) = \max F_n(X^n)\text{, s.t. } X^n \in W_{ni_2}.$$

The process is continued until a local maximum of the problem *B* is reached. In this case $X^{nl*} = \bar{X}^{nm*}$ (fig.2).

Global optimization

Let us assume that each starting point is a random event. This means that the local maximum corresponding to the starting point is a random event as well. Numerical experiments show that random sample histograms of objective function values at local maxima corresponding to random samples of starting points are close to the normal distribution. This assumption enables to use the three sigma rule when looking for an approximation to the global maximum of the problem *B*. The probabilistic properties allow using the DNS to search for an approximation to a global maximum. This probabilistic method is more effective than the Monte-Carlo method [3].

The basic idea of the DNS can be presented by the following simplified scheme. Firstly, a random sample from the set $\Pi \subset R^{2k}$ is realized and an appropriate set of local maxima is computed. Then a starting point, at which the value of the objective function is greatest, is singled out. The point is taken as a centre of a neighbourhood of radius β (diameter of Π), and a random sample from the neighbourhood is performed. After that we compute an appropriate set of local maxima. A starting point corresponding to a local maximum at which the objective function reaches the greatest value is selected. If the value is less than the previous one then the radius decreases and a new random sample from the neighbourhood is realized. If the value is strictly greater than the previous one then a starting point corresponding to the value is taken as a centre of a new neighbourhood of the previous radius and a random sample from the neighbourhood is performed. The computational process is continued until the neighbourhood radius becomes less than given one. A local maximum at which the objective reaches a maximal value is taken as an approximation to a global maximum.

A modification of the DNS consists of the following stages.

I. Initialization stage of the DNS

On this stage we search for promising centres of neighbourhoods for the next stage (*a recurrent stage of the DNS*) as follows.

Firstly, a random sample of vectors from the set Π is generated. On the ground of the vectors starting points are formed and appropriate local maxima are computed. The vectors to which correspond ω the best local maxima are taken as centres of neighbourhoods. In order to choose promising vectors from the set Π for the subsequent iterative process random samples are generated in the neighbourhoods. An arithmetic average and a mean-square distance of the samples are calculated for each neighbourhood.

Step 1. Realize a random sample $\Pi_0 \subset \Pi$ consisting of λ points, construct a set $\mathrm{T}_0 = \left\{X^{nj}, j \in J_\lambda = (1,2,...,\lambda)\right\} \subset \mathrm{T}$ of starting points and form an appropriate set $L_0 = \left\{X^{nj*}, j \in J_\lambda\right\} \subset L$ of local maxima. Thus, to each $p^j \in \Pi_0$ there corresponds a local maximum $X^{nj*} \in L_0$.

Step 2. Choose points $X^{nj_l*} \in L_0$, $l \in I_\omega = \{1,2,...,\omega\}$ so that

$$F_n(X^{nj_1*}) > F_n(X^{nj_2*}) > ... > F_n(X^{nj_\omega*}) \geq \max\left\{F_n(X^{n*}) : X^{n*} \in L_0 \setminus \{X^{nj_l*}, l \in I_\omega\}\right\}.$$

Hence, to each local maximum X^{nj_l*} there corresponds the point $p^{j_l} \in \Pi_0$, $l \in I_\omega$.

Take the points $p^{j_l} \in \Pi_0$ as centres of neighbourhoods $N_{0l} \subset \Pi_0$, $l \in I_\omega$, of radius $\rho^0 < \beta$. Derive random samples $S_{0l} \subset N_{0l}$, $l \in I_\omega$, consisting of λ points. The value $\rho^0 = 0{,}25\beta$ allows to estimate "behaviour" of $F_n(X^n)$ at local maxima close to X^{nj_l*}, $l \in I_\omega$.

Step 3. Form sets $\mathrm{T}_{0l} \subset \mathrm{T}$ and $L_{0l} \subset L$ corresponding $S_{0l} \subset N_{0l}$, $l \in I_\omega$.

Step 4. Compute an arithmetic average m_{0l} and a mean-square distance σ_{0l}, $l \in I_\omega$, of values of $F_n(X^n)$ for each L_{0l}.

Step 5. Determine the point $\widetilde{p}^0$, corresponding to the local maximum $\widetilde{X}^{n0*} = \arg\max\left\{F_n(X^{n*}), \text{s.t. } X^{n*} \in \bigcup_{l=1}^{\omega} L_{0l}\right\}$.

II. Recurrent stage of the DNS

On the basis of the statistical characteristics of random samples of the previous stage new promising centers are found. The choice of the centers is carried out on the ground of assumption about the normal distribution law of objective function values at local maxima. It allows to use the three sigma rule. If after performance of a given number of iterations of the DNS values of objective function at local maxima are not improved then radii of the neighbourhoods fast decreases.

The iterative process begins from $k = 1$.

Let the $(k–1)$-th iteration have been fulfilled. As a result of the iteration a set $L_{(k–1)i}$ is formed, $m_{(k–1)i}$, $\sigma_{(k–l)i}$, $i = 1, 2, 3$, are computed and a point $\widetilde{p}^{k-1}$ corresponding to the local maximum $\widetilde{X}^{n(k-1)*} = \arg\max\left\{F_n(X^{n*}), \text{s.t. } X^{n*} \in \bigcup_{i=1}^{3} L_{(k-1)i}\right\}$, is obtained.

Step 1. Single out centres c^{ki} of neighbourhoods N_{ki}, $i = 1, 2, 3$, the following way:

– $c^{k1} = \begin{cases} \widetilde{p}^{k-1} \text{ if } F_n(\widetilde{X}^{n(k-1)*}) > F_n(\widetilde{X}^{n(k-2)*}), \\ \widetilde{p}^{k-2} \text{ if } F_n(\widetilde{X}^{n(k-1)*}) \le F_n(\widetilde{X}^{n(k-2)*}), \end{cases}$

if $k = 1$, then $c^{k1} = \widetilde{p}^0$;

– $c^{k2} = \begin{cases} c^{(k-1)1} \text{ if } c^{k1} = \widetilde{p}^{k-1} \text{ and } \widetilde{X}^{n(k-1)*} \in L_{(k-1)1}, \\ c^{(k-1)2} \text{ if either } c^{k1} = \widetilde{p}^{k-1} \text{ and } \widetilde{X}^{n(k-1)*} \in L_{(k-2)2}, \text{ or } c^{k1} = \widetilde{p}^{k-2}, \\ c^{(k-1)3} \text{ if } c^{k1} = \widetilde{p}^{k-1} \text{ and } \widetilde{X}^{n(k-1)*} \in L_{(k-1)3}, \end{cases}$

if $k = 1$, then $c^{k2} = p^{j_l}$, where p^{j_l} is a centre of N_{0l} at which $\widetilde{p}^0$ is obtained;

– $c^{k3} = c^{(k-1)i}$, where $c^{(k-1)i}$ is the centre of $N_{(k-1)i}$, at which

$\max\{m_{(k-1)i} + \theta\sigma_{(k-1)i}, i = 1,2,3\}$, $0 < \theta \le 3$ (if $k = 1$, then $i = 1, 2, \ldots, \omega$) is obtained.

Step 2. Define radii ρ_{ki} of the neighbourhoods N_{ki}, $i = 1, 2, 3$, as follows:

$$\rho_{ki} = \begin{cases} \mu\rho_{(k-1)i} \text{ if } m_{(k-1)i} + \theta\sigma_{(k-1)i} \le f^*_{k-1}, \\ \dfrac{1}{\mu}\rho_{(k-1)i} \le \rho_k \text{ if } m_{(k-1)i} + \theta\sigma_{(k-1)i} > f^*_{k-1}, \end{cases}$$

where $f^*_{k-1} = \begin{cases} F_n(\widetilde{X}^{n(k-1)*}) \text{ if } F_n(\widetilde{X}^{n(k-1)*}) > F_n(\widetilde{X}^{n(k-2)*}), \\ F_n(\widetilde{X}^{n(k-2)*}) \text{ if } F_n(\widetilde{X}^{n(k-1)*}) \le F_n(\widetilde{X}^{n(k-2)*}), \end{cases}$

$$\rho_k = \begin{cases} \mu_1\rho_{k-1} \text{ if } F_n(\widetilde{X}^{n(k-1)*}) \le F_n(\widetilde{X}^{n(k-2)*}) \text{ and } F_n(\widetilde{X}^{n(k-1)*}) \le F_n(\widetilde{X}^{n(k-3)*}), \\ \dfrac{1}{\mu_1}\rho_{k-1} \le \beta \text{ if } F_n(\widetilde{X}^{n(k-1)*}) > F_n(\widetilde{X}^{n(k-2)*}), \end{cases}$$

$\mu = 0.8$ is a factor which decreases or increases neighbourhood radii on each stage of the DNS, $\mu_1 = 0.6$ is a factor which guarantees rapid decreasing of neighbourhood radii in the case of absence of an improvement of $\widetilde{X}^{nk*}$. If $k = 1$ then $\rho_k = \rho_{k1} = \rho_{k2} = \rho_{k3} = \beta$.

Step 3. Realize random samples $S_{ki} \subset N_{ki}$, construct sets $T_{ki} \subset T$ and $L_{ki} \subset L$, compute

m_{ki}, σ_{ki}, $i = 1,2,3$, and obtain a point $\tilde{p}^k$.

Step 4. Verify a *stopping criterion.* If the number of identical values of $F_n(X^n)$ on each of L_{ki}, i = 1, 2, 3 is greater than 0.6λ, then we finish the solution process. It should be noted that if the radius value in the DNS is close to the minimal admissible one ($\sqrt{2}r$), then the number of identical local maxima and local maxima, at which the objective has the same value, increases significantly.

Step 5. Set $k \leftarrow k+1$.

The input parameter λ must be chosen carefully since an excessive value of λ leads to a high computational burden. On the other hand, a too small λ results in a solution deterioration. It follows from statistical experiments that the value of λ has to be greater than or equal to 50 to provide a reliability of m_{ki} and σ_{ki}. In additional, a value of λ independs on the problem if $n > 20$.

Computation experiments

All the computational experiments were run on PC with the following main characteristics: Intel Core2Duo E4500 processor and 2 Gb of RAM. The linear programming problems (3)–(4) are calculated by means of HOPDM package (version 2.13) [6]. It should be noted that at present there is a number of modern solvers (for example, Meszaros solver [7], which efficiently solve linear programming problems by the interior point method. The solvers shall allow essentially reducing a runtime.

When looking for approximations to global maxima of problems of kind *B* we take λ = 50, i.e. we compute 50 local maxima into each neighbourhood.

Comparison of results

Since, all known us researches do not solve packing problems of identical circles into an arbitrary multiply connected region then we compare computational results of packing circles into squares and rectangles with the benchmark results presented in [8].

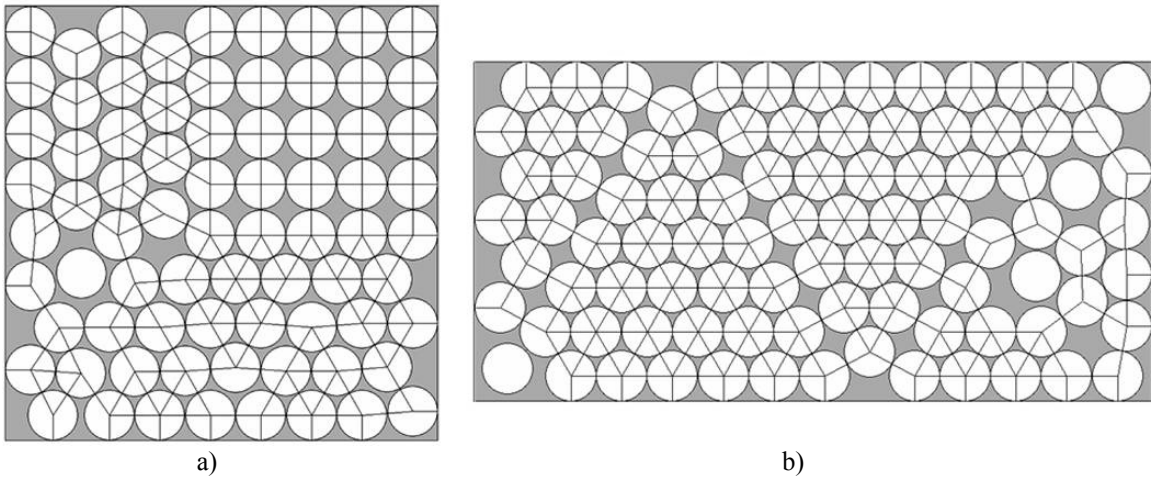
For testing our approach we solve a number of numerical examples of circle packings in the unit square and rectangles presented in the site www.packomania.com. The computational results obtained show that the approach produces the same high performance solutions when using at most 10–30 starting points.

The solution results of the problems 1.1–5.9 in [8] are obtained by our approach as well. Table presents a performance of GENPACK [8] and our approach for first block of problems 1.1–1.9 given in [8] for which we obtain two improvements.

Figure 3 shows improvements of the results of packing problems presented in [8].

Performance of GENPACK and our method for problems given in [8]

Problem			Results				
			Obtained in [10]	obtained by means of GENPACK in [8]		obtained by means of our method	
Name	Box dimensional	Circle radius	Number of packed circles	Number of packed circles	Time, second	Number of packed circles	Time, second
1.1	160×80	6	90	91	734.05	92	5351
1.2	100×200	8	84	84	5791.83	84	4461
1.3	120×240	10	73	74	4065.91	74	3879
1.4	100×80	5	86	86	37108.39	86	9785
1.5	120×80	6	68	68	23.35	68	2311
1.6	120×100	6	87	87	2273.14	87	3456
1.7	80×80	5	68	68	12336.67	68	9874
1.8	100×100	6	70	71	225.57	71	2869
1.9	120×120	7	73	74	18.53	75	3985



a) b)

Fig. 3. Improved results:
a) – the result of the packing problem of circles of radius 7 in the square 120×120;
b) – the result of the packing problem of circles of radius 6 in the rectangle 160×80.

Figure 3,a illustrates the packing problem of circles of radius 7 in the square 120×120 (the problem 1.9 in [8]). The result in [8] is improved by one circle.

Figure 3,b illustrates the packing problem of circles of radius 6 in the rectangle 160×80 (the problem 1.1 in [8]). The result in [8] is improved by one circle.

Numerical examples

Abilities of the mathematical model offered and solution methods are demonstrated by a choice of test instances.

Let there be a set of identical circles and a multiply connected region P presented in fig. 1, a (given in [1]). The frontier of P_0 (fig. 1, b) (given in [1]) is given by the sequence of the following line segments and arcs of circles: $[x_1, y_1, x_2, y_2]$, $[x_2, y_2, x_3, y_3, \bar{x}_1, \bar{y}_1, \bar{r}_1]$, $[x_3, y_3, x_4, y_4]$, $[x_4, y_4, x_5, y_5]$, $[x_5, y_5, x_6, y_6]$, $[x_6, y_6, x_7, y_7, \bar{x}_2, \bar{y}_2, \bar{r}_2]$, $[x_7, y_7, x_8, y_8, \bar{x}_3, \bar{y}_3, \bar{r}_3]$, $[x_8, y_8, x_9, y_9]$, $[x_9, y_9, x_{10}, y_{10}]$, $[x_{10}, y_{10}, x_1, y_1]$, where $(x_1, y_1) = (10, 60)$, $(x_2, y_2) = (17.6, 75.19)$, $(x_3, y_3) = (28.15, 77.41)$, $(x_4, y_4) = (30, 70)$, $(x_5, y_5) = (60, 80)$, $(x_6, y_6) = (62.88, 70)$, $(x_7, y_7) = (80.26, 39.84)$, $(x_8, y_8) = (62.34, 30.95)$, $(x_9, y_9) = (45, 35)$, $(x_{10}, y_{10}) = (28, 25)$, $(\bar{x}_1, \bar{y}_1) = (20, 90)$, $(\bar{x}_2, \bar{y}_2) = (63, 50)$, $(\bar{x}_3, \bar{y}_3) = (75, 28)$, $\bar{r}_1 = 15$, $\bar{r}_2 = 20$, $\bar{r}_3 = 13$.

The prohibited areas are represented as follows:

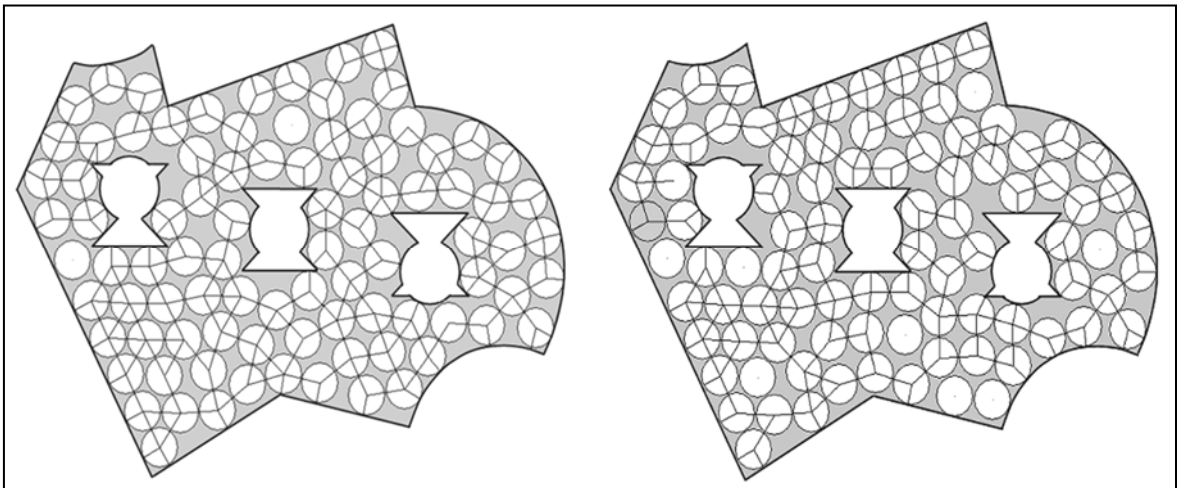
$$A_l = C_{l1} \bigcup \left(\bigcup_{q=1}^{2} M_{lq} \right), \quad l \in I_\sigma = \{1,2,3\}, \tag{5}$$

where $C_{11} = \{(x, y) \in R^2 : (x - 45)^2 + (y - 55)^2 - 4^2 \le 0\}$,

$C_{12} = \{(x, y) \in R^2 : (x - 25)^2 + (y - 65)^2 - 4^2 \le 0\}$,

$C_{13} = \{(x, y) \in R^2 : (x - 65)^2 + (y - 50)^2 - 4^2 \le 0\}$, M_{11}, M_{12}, M_{21}, M_{22}, M_{31} and M_{32} are the triangles given by their vertex coordinates: (45, 55), (50, 50) and (40, 50); (45, 55), (40, 60) and (50, 60); (25, 58), (30, 53) and (20, 52); (25, 58), (20, 63) and (30, 63); (65, 52), (70, 47) and (60, 47); (65, 52), (60, 57) and (70, 57) respectively.

In the paper we present result of packing circle into given region P if $r = 2.5$ (fig. 4). Figure 4, a illustrates the packing of 90 circles of radius 2.5 corresponding to an approximation to a global maximum of the problem A, and fig. 4, b illustrates the packing of circles corresponding to an approximation to a global maximum X^{n*} of the problem B when packing $n = 91$ circles. The ra-



a) b)

Fig. 4. Results of packings if r = 2.5:
a) – the result of packing of 90 circles; b) – the result of packing of 91 circles

dius of the black-out circle in fig. 4, b is equal to 2.45, i. e. 91 circles of radius 2.5 can not be packed into P.

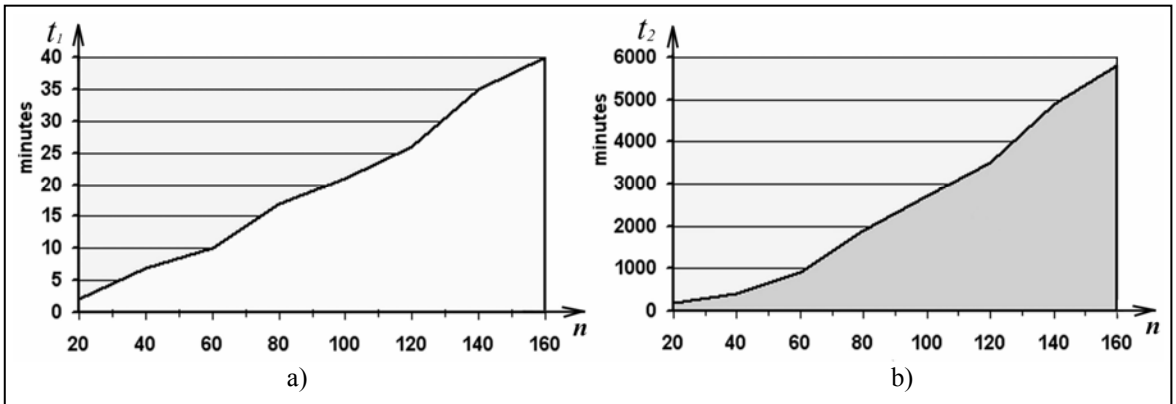
Dependences of the runtime on the number of circles being packed are shown in fig. 5. The graphic depicted in fig. 5, a shows the runtime t_1 which is expended to pack n circles. It should be noted that the runtime t_1 strongly depends on the estimation of n_0 of circles that can be packed into P. Figure 5, b shows the runtime t_2 which is expended to solve the problem B and to prove that $n + 1$ circles can not be packed into P.

It follows from the graphics shown in fig. 5, b that if $n > 60$, then the runtime essentially increases. In this case an application of the DNS demands a paralleling of the computational process.

Conclusions

This work presents a solution method to solve the problem of packing identical circles into a multiply connected region. The computational results demonstrate that the proposed approach produces high performance solutions. The comparison of obtained results with the world analogues shows that the approach is promising.

We offer an efficient methodology to reduce the computational burden when searching for local maxima. It should be noted that the basic part of runtime is expended to prove that no more circle can be packed. The runtime depends strongly on a complexity of the placement region shape (i.e. on the number of prohibited areas and the primary objects of type Q_{j2} and Q_{j4} [1]) that form the region P).



a) b)

Fig. 5. Dependences of the runtime on the number of circles:
a) – the runtime t_1; b) – the runtime t_2