A. Pankratov, D. Sc.
T. Romanova, D. Sc.
A. Kotelevskiy

Institute for Mechanical Engineering Problems of the National Academy of Sciences of Ukraine,
Kharkov, Ukraine,
e-mail: impankratov@mail.ru

Ключові слова: *задача упаковки, довільні об'єкти, неперервні обертання, допустимі відстані, phi-функції, генератор простору розв'язків, вирішувач, нелінійна оптимізація.*

УДК 519.85

LAYOUT PROBLEMS FOR ARC OBJECTS IN CONVEX DOMAINS

Розглянуто оптимізаційну задачу упаковки довільних об'єктів, обмежених дугами кіл та відрізками прямих в опуклі області. Побудовано математичну модель у вигляді задачі недиференційованої оптимізації, множина реалізацій якої покриває широкий клас наукових і прикладних задач геометричного проектування. Розроблено методологію розв'язання задач упаковки з урахуванням технологічних обмежень (мінімально допустимі відстані, зони заборони, можливість неперервних трансляцій та обертань об'єктів). Запропоновано генератор простору розв'язків та вирішувач (solver) для автоматичного розв'язання NLP-задач розглянутого класу.

Introduction

The layout problem is a part of computational geometry that has rich applications in garment industry, sheet metal cutting, furniture making, shoe manufacturing, glass industry shipbuilding industry, etc. The common task in these areas is to arrange a set of shapes of specified shapes and sizes within a given sheet (strip) of material (textile, wood, metal, glass etc.) [1, 2]. To minimize waste one wants to arrange shapes as close to each other as possible.

The problems are NP-hard [3], and as a result solution methodologies predominantly utilize heuristics and nearly all practical algorithms deal with shapes which are approximated by polygons (see tutorials [4, 5] and references therein). The most popular and most frequently cited tool in the modern literature on the Cutting and Packing is the No-Fit Polygon, it is designed to work for polygonal objects without rotations. A notable exception being [6–9], which allows circular shapes, but they cannot be freely rotated. Tools of packing of rotated polygons is considered in [10, 11]. Paper [12] is mainly focused on presenting and discussing efficient tools and representations to tackle the geometric layer of layout algorithms that capture the needs of the real-world applications of irregular packing problems. In [13] an extended local search algorithm (ELS) for the irregular strip packing problem is discussed. Objects are approximated by polygons and can be free rotated. It adopts two neighborhoods, swapping two given polygons in a placement and placing one polygon into a new position. The local search algorithm is used to minimize the overlap on the basis of the neighborhoods mentioned above and the unconstrained nonlinear programming model is adopted to further minimize the overlap during the search process. Moreover, the tabu search algorithm is used to avoid local minima, and a compact algorithm is presented to improve the result. The results of standard test instances indicate that when compared with other existing algorithms, the presented algorithm does not only show some signs of competitive power but also updates several best known results.

Due to the extreme complexity of the analytical description of the relationship between geometric objects, bounded by circular arcs and lines segments, only a few papers devoted to placement of arbitrary shaped objects.

We present the layout problems in a formal mathematical manner. In the paper we deal with objects of very general shape and we characterize their arrangements by means of special phi-functions [14–16]. As a convex domain Ω we consider a nonempty intersection of finite number of convex polygons and circles (in particular: a rectangle, a convex polygon and a circle).

The concept the phi-functions is a highly convenient for practical solution of the layout problem. In particular, we take advantage of phi-functions to develop more efficient algorithms.

Our principal goal is to present here a generator of mathematical models of layout problems using the phi-function technique and demonstrate practical benefits of our algorithms and NLP-solver for non-smooth layout problems.

We consider layout problems in the following basic formulation**:**

© A. Pankratov, T. Romanova, A. Kotelevskiy, 2016

Basic layout problem. Place a set of objects T_i, $i \in \{1, 2, \ldots, N\} = I_N$ within a convex domain Ω of variable metrical characteristics p, so that the given restrictions on the placement of the objects are fulfilled and the area of Ω reaches the minimal value.

We assume, that each item T_i is two-dimensional phi-object [14] (as a model of real objects), bounded by line segments and circular arcs (see appendix A for details of definition of placement objects). We allow here free rotations and translations of objects. The restrictions include: containment of objects into a container, non-overlapping of objects, given minimal allowable distances between objects, prohibited areas, rotation constraints, and other specific technological restrictions (e. g. a given allowable ranges of rotation angles).

A multiplicity of shapes of $T_i \subset R^2$ as well as a variety of restrictions creates a wide spectrum of subsequent problems of the *basic layout problem*. Our intention is to present each of the subsequent problems as a nonlinear programming problem. To this aim we provide a generation of a solution space for the class of problems based on phi-functions technique. Using the mathematical model generator we develop efficient optimisation algorithm for solving layout problems.

Mathematical model and its properties

We assume that any placement object T (an object which has to be placed into a container) considered here, is a two-dimensional phi-object, bounded by line segments, convex and concave circular arcs [17]. The location and orientation of a *placement* object T is defined by a variable vector of its placement parameters $u_T = (x_T, y_T, \theta_T)$. The translation of object T by vector $v_T(x_T, y_T) \in R^2$ and the rotation of T (with respect to its reference point) by angle θ_T is defined by $T(v_T, \theta_T) = \{t \in R^2 : t = v_T + M(\theta_T)t^0, \forall t^0 \in T^0\}$, where T^0 denotes the non-translated and non-rotated object T, $M(\theta_T)$ is rotation matrix.

We assume here that placement objects have fixed sizes (metrical characteristics).

Let $u = (p, u_1, u_2, \ldots, u_N) \in R^\sigma$ is a vector of variables, where $(u_1, u_2, \ldots, u_N)$ is a vector of variable placement parameters (motion vector) $u_i = (x_i, y_i, \theta_i) = (v_i, \theta_i)$ of T_i, $i \in I_N$, R^σ is the arithmetic Euclidean space of σ-dimension, $(x_i, y_i) \in R^2$ is a translation vector and θ_i is a rotation parameter of T_i.

Mathematical model of the basic layout problem may have the form:

$$\min \kappa(u), \quad \text{s. t. } u \in W \subset R^\sigma, \tag{1}$$

$$W = \{u \in R^\sigma : \Phi'_\tau \geq 0, \Phi''_i \geq 0, \varphi_t \geq 0, \tau = 1,2,\ldots,\lambda, i = 1,2,\ldots,N, t = 1,2\ldots,M\}, \tag{2}$$

where $\kappa(u)$ is an area of Ω;

function Φ'_τ is a *phi-function* Φ^{AB} (see e. g. [9]) for describing *non-overlapping* constraint $\mathrm{int}A \cap \mathrm{int}B = \varnothing$ of objects A and B, or *adjusted phi-function* $\widehat{\Phi}^{AB}$ (see e.g. [9]) for describing *distance* constraint $\mathrm{dist}(A, B) \geq \rho$ ($\mathrm{int}(A \oplus C(\rho)) \cap \mathrm{int}B = \varnothing$), here $\mathrm{dist}(A, B) = \min\limits_{a \in A, b \in B} d(a, b)$, $d(a, b)$ is the Euclidean distance between points a and b, ρ is a given minimal allowable distance between objects A and B. Here $C(\rho)$is a circle of radius ρ, int(•)is the interior of object (•) [18], $\oplus$ is the symbol of Minkovski sum, $\lambda = 0{,}5N(N - 1)$;

function Φ''_i is a *phi-function* $\Phi^{A\Omega^*}$ for describing *containment* constraint $A \subset \Omega \Leftrightarrow \mathrm{int}A \cap \Omega^* = \varnothing$, $\Omega^* = R^2 \backslash \mathrm{int}\Omega$, or *adjusted phi-function* $\widehat{\Phi}^{A\Omega^*}$ for describing *distance* constraint between objects A and Ω^*, i. e. $\mathrm{dist}(A, \Omega^*) \geq \rho$ ($A \oplus C(\rho) \subset B \Leftrightarrow \mathrm{int}A \cap (B^* \oplus C(\rho)) = \varnothing$), where ρ is a given minimal allowable distance between objects A and Ω^*;

$\varphi_t \geq 0$, $t = 1, 2, \ldots, M$, is a system of additional restrictions on values of components of vector u (e. g. a given ranges of translation variables or rotation angles) if any, provided that each function φ_t is smooth.

We would remind the reader that phi-functions are continuous and everywhere defined functions which allow us to describe analytically relations between two arbitrary shaped phi-objects A and B in such a way: a) $\Phi^{AB} > 0$ if $\mathrm{int}A \cap \mathrm{int}B = \varnothing$, b) $\Phi^{AB} = 0$ if $\mathrm{int}A \cap \mathrm{int}B = \varnothing$ and $\mathrm{fr}A \cap \mathrm{fr}B \neq \varnothing$, c) $\Phi^{AB} < 0$ if $\mathrm{int}A \cap \mathrm{int}B \neq \varnothing$. Here Φ^{AB} means phi-functions for phi-objects A and B, fr(•)means the frontier of object fr(•).

By definition an adjusted *phi*-function is an everywhere defined continuous function $\widehat{\Phi}^{AB}$ of objects A and B, such as: a) $\widehat{\Phi}^{AB} > 0$, if dist(A, B) > ρ, b) $\widehat{\Phi}^{AB} = 0$, if dist(A, B) = ρ, c) $\widehat{\Phi}^{AB} < 0$, if dist(A, B) < ρ. In particular, we have dist(A, B) ≥ ρ $\Leftrightarrow$ $\widehat{\Phi}^{AB} \geq 0$.

We emphasize that each *phi*-function (or *adjusted* phi-function) for a pair of 2D *phi*-objects is radical-free piecewise continuously differentiable function (see e. g. [16]), and defined in terms of operations of maximum or/and minimum of smooth functions.

In [15] has been proved that objects A and B made by line segments and circular arcs can be always presented as a finite union of basic objects. We refer the reader to the paper for details of the definition of set $\mathfrak{R}$ of basic objects and the algorithm of decomposition of arbitrary shaped objects by basic ones.

Let objects $A = \bigcup_{i=1}^{n_A} A_i$ and $B = \bigcup_{j=1}^{n_B} B_j$ be given, A_i, $B_j \in \mathfrak{R}$. As it is known [14, 15], phi-function for the pair of objects A and B has the form

$$\Phi^{AB} = \min\{\Phi_{ij},\ i = 1,\ \dots,\ n_A,\ j = 1,\ \dots,\ n_B\}. \tag{3}$$

Here Φ_{ij} is a phi-function for a pair of basic objects A_i and B_j. We further call the phi-function as a *basic phi-function*. We present relation (3) as follows

$$\Phi^{AB} = \min\{\Phi_k,\ k = 1,\ \dots,\ n_A \cdot n_B\}. \tag{4}$$

where Φ_k is a basic phi-function.

Using (4) let us introduce the following function

$$\Upsilon(u) = \min\{\Phi'_\tau, \tau = 1,2,...,\lambda, \Phi''_i, i = 1,2,...,N, \varphi_t, t = 1,2...,M\}, \tag{5}$$

which we call an *arrangement* function. This function is piecewise continuously differentiable function and depends on all variables (p, u_1, u_2, …, u_N) of problem (1)–(2). We note that $\Upsilon(u) \geq 0$ if and only if $\Phi'_\tau \geq 0$ for all $\tau = 1, 2, \dots, \lambda$, and $\Phi''_i \geq 0$ for all $i = 1, 2, \dots, N$ and $\varphi_t \geq 0$, for all $t = 1, 2, \dots, M$.

Let us denote by Φ_k, $k = 1, 2, \dots, n\, k = 1, .2, .., n$, all basic phi-functions in (5) $n = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} n_i \cdot n_j + \sum_{i=1}^{N} n_i$, where $\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} n_i \cdot n_j$ is the number of all basic phi-functions for non-overlapping constraints; $\sum_{i=1}^{N} n_i$ is the number of all basic phi-functions for containment constraints. Here n_i is the number of basic objects forming composed object T_i and n_j is the number of basic objects forming composed object T_j.

Then relation (5) can be defined as

$$\Upsilon(u) = \min\{\Phi_k,\ k = 1,\ \dots,\ n,\ \varphi_t,\ t = 1,\ 2,\ \dots,\ M\}, \tag{6}$$

Now (2) may be presented in the equivalent form:

$$W = \{u \in R^\sigma : \Upsilon(u) \geq 0\}, \tag{7}$$

where function $\Upsilon(u)$ has form (6).

Let us consider general characteristics of problem (1)–(2).

1) Due to phi-functions (adjusted phi-functions) the solution space W given by (7) can be represented as

$$W = \bigcup_{s=1}^{\eta} W_s, \tag{8}$$

where $W_s = \{u \in R^\sigma : \Upsilon_s(u) \geq 0\}$,

$$\Upsilon_s = \min\{f_s^i, i \in I_s\}, \qquad f_s^i \in \{f\}, \tag{9}$$

hereafter $\{f\}$ notes a family of continuously differentiable functions; inequality $\Upsilon_s \geq 0$ is equivalent to a system of inequalities $f_s^i \geq 0$, $f_s^i \in \{f\}$, $i \in I_s$. In (8) $\eta \leq \eta^*$, η^* is the upper estimation of η.

2) Optimisation problem (1)–(2) is NP-hard nolinear programming problem with nonsmooth functions

3) The solution space W has a complicated structure: it is, in general, a disconnected set, each connected component of W is multiconnected, the frontier of W is made of nonlinear surfaces containing valleys, ravines, etc.

4) Based on (8), problem (1)–(2) can be reduced to the following optimisation problem:

$$\kappa(u^*) = \min\{\kappa(u_1^*),...,\kappa(u_s^*),...,\kappa(u_\eta^*)\}, \tag{10}$$

where

$$\kappa(u_s^*) = \min_{u\in W_s \subset R^\sigma} \kappa(u), \qquad s = 1,2,...,\eta \le \eta^*. \tag{11}$$

Clearly, the global solution can be obtained and proved by inspecting and exactly solving all of the subproblems defined in (11). Subproblems (11) are in general nonlinear programming problems and they may be solved by standard techniques (e. g. interior point method, feasible direction method) of local optimisation.

Our goal is to create a generator of solution space W defined by (7), which results in an automatic generation of solution subregion W_s for subproblems (11), s = 1, 2, …, η.

To this aim we transform functionϒ, which is defined by (5) to equivalent formula

$$\Upsilon(u) = \max\{\Upsilon_s,\ s = 1, 2, \ldots, \eta\}, \tag{12}$$

where Υ_s is given by (9).

Such transformation is always possible. It follows from algebra of logic formulas. We offer here a way of construction of function (12) based on, so called, solution tree.

Solution tree

We desire to describe the solution space W defined by (7) using, so called, *solution tree* $\Im^\Upsilon$**,** such that each terminal node υ_s of the tree corresponds to inequality $\Upsilon_s \ge 0$, s = 1, 2, …, η. Here function Υ_s is defined by (9).

Inequality tree (I-tree)

Let F be a piecewise continuously differentiable function and formed by operations of maximum and minimum of functions from $\{f\}$. We introduce, so-called, inequality tree (hereafter I-tree) for describing inequality $F \ge 0$.

We construct I-tree in the following manner. Let v_1^0 be the root node of $\Im$ associated with $F \ge 0$; v_k^l be k-th node of l-th level of $\Im$**,** l = 0, 1, …, L. We associate with node v_k^l of I-tree $\Im$ inequality $F_k^l \ge 0$ (for the sake of simplicity, we further say, simply F_k^l). Each function F_k^l may take two forms: $F_k^{\circ l} = \max\{F_{kj}, j = 1,...,N_k^l\}$ or $F_k^{\bullet l} = \min\{F_{kj}, j = 1,...,N_k^l\}$. We say that v_k^l is an *additive* node if it corresponds to $F_k^{\circ l}$, and v_k^l is a *multiplicative* node if it corresponds to $F_k^{\bullet l}$. Node v_k^l is a *terminal* if: 1) v_k^l is a *multiplicative* one and 2) all functions $F_{kj}, j = 1,...,N_k^l$, belong to $\{f\}$. If v_k^l is not a *terminal* node, then $F_{kj} = F_{kj}^{l+1}, j = 1,...,N_k^l$, where N_k^l is the number of child nodes of (l + 1)-th level of $\Im$ generating by node v_k^l.

Let us consider examples of constructing I-tree $\Im$ for two functions

$$F_1 = \min\{\max\{f_1, f_2\}, \min\{f_3, f_4\}, \max\{f_5, f_6\}\}, \text{ and}$$

$$F_1 = \max\{\max\{f_1, f_2\}, \min\{f_3, f_4\}, \max\{f_5, f_6\}\}$$

where $f_i \in \{f\}$, $i \in I_6$.

We form the I-trees for $F_1 \ge 0$ and $F_2 \ge 0$ as follows.

We set root node of each I-tree: $F_1^{\bullet 0} = F_1$ is a *multiplicative* node for $F_1 \ge 0$ (Fig. 1, a) and $F_1^{\circ 0} = F_2$ is an *additive* node for $F_2 \ge 0$ (Fig. 2, a) respectively.

The first level of both I-trees is the same and contains three nodes: $F_1^{\circ 1} = \max\{f_1, f_2\}$ is an *additive*, $F_2^{\bullet 1} = \min\{f_3, f_4\}$ is a *multiplicative*, $F_3^{\circ 1} = \max\{f_5, f_6\}$ is an *additive*.

The second level of both I-trees is the same and contains five nodes: $F_{11}^{\bullet 2} = f_1$, $F_{12}^{\bullet 2} = f_2$, $F_{21}^{\bullet 2} = F_2^{\bullet 1} = \min\{f_3, f_4\}$, $F_{31}^{\bullet 2} = f_5$, $F_{32}^{\bullet 2} = f_6$.

Each node of the level is a *multiplicative*.

I-trees $\Im_1$ and $\Im_2$ associated with $F_1 \geq 0$ and $F_2 \geq 0$ are given in Fig. 1, a and Fig. 2, a.

Transformation of I-tree $\Im$ into I-tree $\tilde{\Im}$

Our aim now to transform multi-level I-tree $\Im$ associated with $F \geq 0$ into the two-level I-tree $\tilde{\Im}$ associated with $\tilde{F}^{\circ} \geq 0$, where

$$\tilde{F}^{\circ} = \max\{F_1^{\bullet}, ..., F_s^{\bullet}, ..., F_m^{\bullet}\}, \qquad F_s^{\bullet} = \min\{f_s^i, i \in I_s\}, \qquad f_s^i \in \{f\}.$$

Terminal nodes of I-tree $\tilde{\Im}$ have to correspond to a system of inequalities $f_j^i \geq 0,\ i \in I_j$, $f_j^i \in \{f\}$. We denote the number of inequality systems with smooth functions generated by root node v_1^0 by $\eta^0 = \eta_1^0$.

Each node v_k^l *per se* may be considered as a root node of some sub-tree $\Im_k^l$ of I-tree $\Im$.

We derive the number of inequality systems of $\Im_k^l$ with smooth functions generated by node v_k^l in the form:

$$\eta_k^l = \sum_{j=1}^{N_k^l} \eta_{kj}^{l+1} \text{ if } v_k^l \text{ is } \textit{additive} \text{ node for } l = 1, \ldots, L-1, \qquad (13)$$

$$\eta_k^l = \prod_{j=1}^{N_k^l} \eta_{kj}^{l+1} \text{ if } v_k^l \text{ is } \textit{multiplicative} \text{ node for } l = 1, \ldots, L-1. \qquad (14)$$

We turn to functions of our previous example in order to explain the way of constructing I-tree $\tilde{\Im}$. We show now a way of transformations of I-trees $\Im_1$ and $\Im_2$ for with $F_1 \geq 0$ and $F_2 \geq 0$ in order to get I-trees $\tilde{\Im}_1$ and $\tilde{\Im}_2$ for $\tilde{F}_1^{\circ} \geq 0$ and $\tilde{F}_2^{\circ} \geq 0$, where

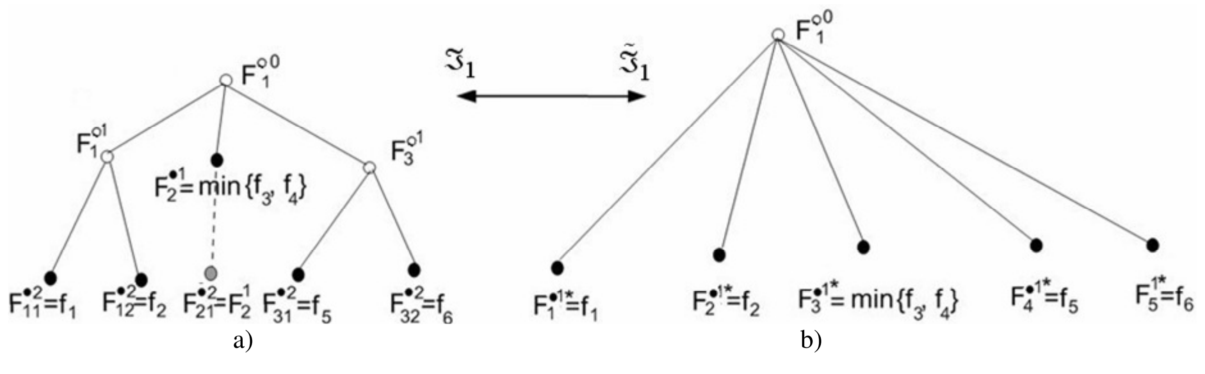
$\tilde{F}_1^{\circ} = \max\{\min\{f_1, f_3, f_4, f_5\}, \min\{f_1, f_3, f_4, f_6\}, \min\{f_2, f_3, f_4, f_5\}, \min\{f_2, f_3, f_4, f_6\}\}$

$\tilde{F}_2^{\circ} = \max\{f_1, f_2, \min\{f_3, f_4\}, f_5, f_6\}$.

I-trees $\tilde{\Im}_1$ and $\tilde{\Im}_2$ associated with $\tilde{F}_1^{\circ} \geq 0$ and $\tilde{F}_2^{\circ} \geq 0$ are given in Fig. 1, b and 2, b.

The collection of systems $\{\Upsilon_s \geq 0,\ s = 1, \ldots, \eta^0\}$ results from the type of the root node in the following sense.

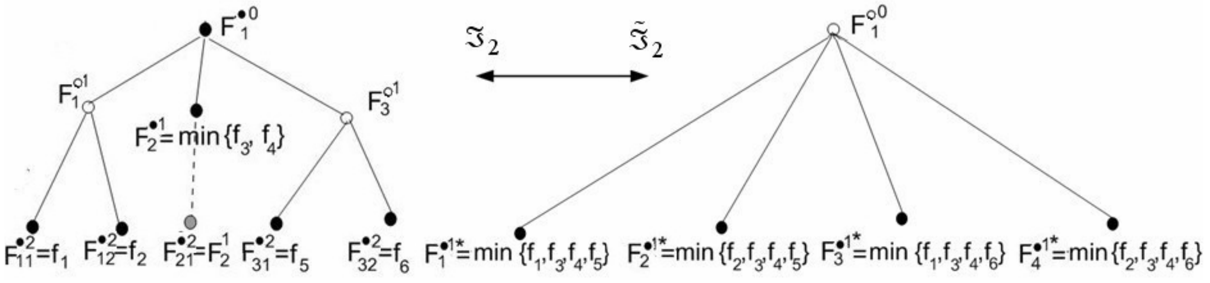
If v_1^0 is a *additive* node then the collection $\{\Upsilon_s \geq 0,\ s = 1, \ldots, \eta^0\}$, for all terminal nodes of $\tilde{\Im}_1$ directly coincides with the collection of systems corresponding to the terminal nodes of $\Im_1$, (in our examples



a) b)

Fig. 1. I-trees:

a) – $\Im_1$ for $F_1 \geq 0$; b) – $\tilde{\Im}_1$ for $\tilde{F}_1^{\circ} \geq 0$


$F_1^{\bullet 0}$
$F_1^{\circ 1}$
$F_2^{\bullet 1}= \min\{f_3, f_4\}$
$F_3^{\circ 1}$
$F_{11}^{\bullet 2}=f_1$ $F_{12}^{\bullet 2}=f_2$ $F_{21}^{\bullet 2}=F_2^1$ $F_{31}^{\bullet 2}=f_5$ $F_{32}^{\bullet 2}=f_6$
$\Im_2$ $\tilde{\Im}_2$
$F_1^{\circ 0}$
$F_1^{\bullet 1*}=\min\{f_1,f_3,f_4,f_5\}$ $F_2^{\bullet 1*}=\min\{f_2,f_3,f_4,f_5\}$ $F_3^{\bullet 1*}=\min\{f_1,f_3,f_4,f_6\}$ $F_4^{\bullet 1*}=\min\{f_2,f_3,f_4,f_6\}$

a) b)

Fig. 2. I-trees:

a) – $\Im_2$ for $F_2 \geq 0$; b) – $\tilde{\Im}_2$ for $\tilde{F}_2^{\circ} \geq 0$

$\eta = 5$, see Fig. 1), I .e.

$$\Upsilon_1 \geq 0 \Rightarrow f_1 \geq 0, \qquad \Upsilon_2 \geq 0 \Rightarrow f_2 \geq 0, \qquad \Upsilon_3 \geq 0 \Rightarrow \{f_3 \geq 0, f_4 \geq 0, \qquad \Upsilon_4 \geq 0 \Rightarrow f_5 \geq 0,$$

$$\Upsilon_4 \geq 0 \Rightarrow f_6 \geq 0, \qquad \eta^0 = \eta_1^0 = \sum_{k=1}^{3} \eta_k^1 = 5, \qquad \eta_1^1 = 2, \qquad \eta_2^1 = 1, \qquad \eta_3^1 = 2.$$

If v_1^0 is a *multiplicative* node then each system $\Upsilon_s \geq 0$ corresponding to terminal node of $\tilde{\Im}_2$ (in our examples $\eta = 4$, see Fig. 2) involves one of inequalities generating by each additive node (for our example: $F_1^{\circ 1} = \max\{F_{11}^2, F_{12}^2\} \geq 0$ and $F_3^{\circ 1} = \max\{F_{31}^2, F_{32}^2\} \geq 0$, see Fig. 2, a) and all inequalities generating by *multiplicative* nodes (for our example: $F_2^{\bullet 1} = \min\{f_3, f_4\} \geq 0$, see Fig. 2, a), i. e.

, ,

$$\Upsilon_1 \geq 0 \Rightarrow \{f_1 \geq 0, f_3 \geq 0, f_4 \geq 0, f_5 \geq 0, \qquad \Upsilon_2 \geq 0 \Rightarrow \{f_1 \geq 0, f_3 \geq 0, f_4 \geq 0, f_6 \geq 0,$$

$$\Upsilon_3 \geq 0 \Rightarrow \{f_2 \geq 0, f_3 \geq 0, f_4 \geq 0, f_5 \geq 0, \qquad \Upsilon_4 \geq 0 \Rightarrow \{f_2 \geq 0, f_3 \geq 0, f_4 \geq 0, f_6 \geq 0,, \quad ,$$

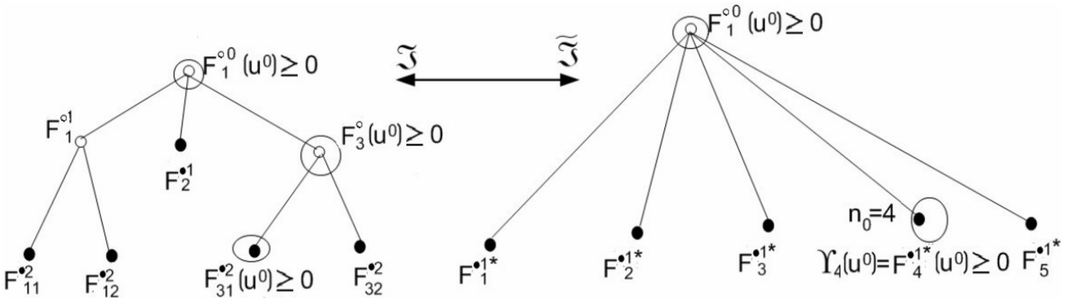
$$\eta^0 = \eta_1^0 = \prod_{k=1}^{3} \eta_k^1 = 4, \qquad \eta_1^1 = 2, \qquad \eta_2^1 = 1, \qquad \eta_3^1 = 2.$$

Multiplicative nodes of I-trees are shown as filled circles, and *additive* nodes of I-trees are shown as empty circles in Fig. 1–4.

We further call I-tree for phi-function a *phi-tree*. Based on transformation of *phi*-tree given above we always may present a basic phi-function Φ_k in (6) as follows

$$\Phi_k = \max_{i=1,\ldots,\eta_k} f_i^k = \max_{i=1,\ldots,\eta_k} \min_{j=1,\ldots,J_i^k} f_{ij}^k, \tag{15}$$

where $f_{ij}^k \in \{f\}$.


$F_1^{\circ 0}(u^0) \geq 0$
$F_1^{\circ 1}$
$F_2^{\bullet 1}$
$F_3^{\circ}(u^0) \geq 0$
$F_{11}^{\bullet 2}$ $F_{12}^{\bullet 2}$ $F_{31}^{\bullet 2}(u^0) \geq 0$ $F_{32}^{\bullet 2}$
$\Im$ $\tilde{\Im}$
$F_1^{\circ 0}(u^0) \geq 0$
$n_0=4$
$F_1^{\bullet 1*}$ $F_2^{\bullet 1*}$ $F_3^{\bullet 1*}$ $\Upsilon_4(u^0)=F_4^{\bullet 1*}(u^0) \geq 0$ $F_5^{\bullet 1*}$

a) b)

Fig. 3. I-trees:

a) – $\Im_1$ for $F_1(u^0) \geq 0$; b) – $\tilde{\Im}_1$ for $\tilde{F}_1(u^0) \geq 0$

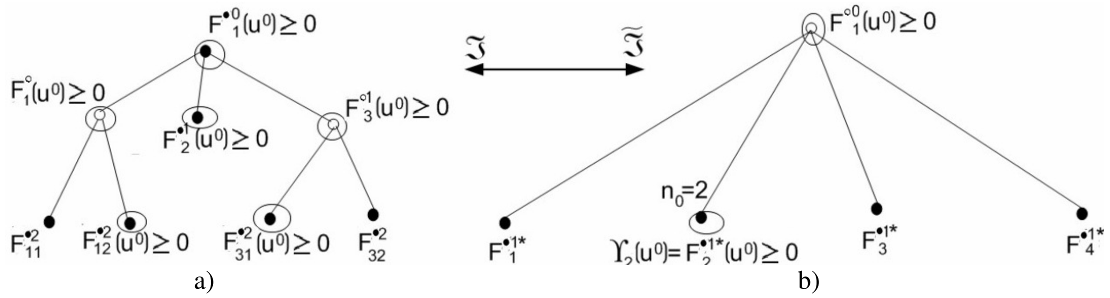

$F^{\bullet 0}_1(u^0) \geq 0$
$F^{\circ}_1(u^0) \geq 0$
$F^{\bullet 1}_2(u^0) \geq 0$
$F^{\circ 1}_3(u^0) \geq 0$
$F^{\bullet 2}_{11}$
$F^{\bullet 2}_{12}(u^0) \geq 0$
$F^{\bullet 2}_{31}(u^0) \geq 0$
$F^{\bullet 2}_{32}$
a)
$\Im$
$\tilde{\Im}$
$F^{\circ 0}_1(u^0) \geq 0$
$n_0=2$
$F^{\bullet 1*}_1$
$\Upsilon_2(u^0)= F^{\bullet 1*}_2(u^0) \geq 0$
$F^{\bullet 1*}_3$
$F^{\bullet 1*}_4$
b)

Fig. 4. I-trees:
a) – $\Im_2$ for $F_2(u^0) \geq 0$; b) – $\tilde{\Im}_2$ for $\tilde{F}_2(u^0) \geq 0$

Finding number n^0 of a system which corresponds to the terminal node of the transformed tree $\tilde{\Im}$ and associated with $F(u^0) \geq 0$

Now we show the way to find a system $\Upsilon_{n^0} = \Upsilon_{n^0}(u^0) \geq 0$ and an appropriate number $n^0 \in \{1,...,\eta_1^0\}$ of the terminal node of the transformed I-tree $\tilde{\Im}$ without its direct construction. Let u^0 be given, so that $F(u^0) \geq 0$.

We find a number n^0 of the terminal node of the transformed tree $\tilde{\Im}$ which associated with $F(u^0) \geq 0$ in the following manner.

We denote a number of the terminal node of the transformed I-tree for $\Upsilon_{n^0} \geq 0$ by n^0.

Let us consider two cases: a) root node v_1^0 is *additive* and b) root node v_1^0 is *multiplicative*:

1) if root node v_1^0 is a *additive* and $F(u^0) \geq 0$ then one of the systems $F_k^1(u) \geq 0$, $k \in \{1,2,...,N_1^0\}$, has to be fulfilled at point u^0.

Now we consider the sub-tree $\Im^k$ with the root node v_k^1. Since $F_k^1(u^0) \geq 0$ then we are sure that there exists some number n_k of terminal node of the tranformed sub-tree $\tilde{\Im}^k$ associated with inequality $\Upsilon_{n_k}^k(u^0) \geq 0$.

Number n_1^0 of corresponding terminal node of the transformed I-tree $\tilde{\Im}$ is derived by recursive formula:

$$n^0 = n_1^0 = \begin{cases} n_k, \text{if } k = 1 \\ n_k + \sum_{i=1}^{k-1} \eta_i^1, \text{if } k \in \{2,...,N_1^0\}. \end{cases} \tag{16}$$

Formula (16) for node v_t^l, $0 \leq l < L - 1$, takes the form:

$$n_t^l = \begin{cases} n_{tk}, \text{if } k = 1 \\ n_{tk} + \sum_{i=1}^{k-1} \eta_i^{l+1}, \text{if } k \in \{2,...,N_1^l\}, \end{cases} \tag{17}$$

subject to $F_{tk}^{l+1}(u_0) \geq 0$.

For our example shown in Fig. 3, a, applying (16), we have:

$$k = 3, \qquad n_3 = 1, \qquad \eta_1 = 2, \qquad \eta_2 = 1, \qquad n^0 = n_3 + \sum_{i=1}^{3-1} \eta_i^1 = n_3 + \eta_1^1 + \eta_2^1 = 1 + 2 + 1 = 4.$$

2) if root node v_1^0 is a *multiplicative* one and $F(u^0) \geq 0$ then $F_k^1(u^0) \geq 0$, for all $k = 1,2,...,N_1^0$.

Thus for each sub-tree $\Im_k^1$ with root node v_k^1, $k \in \{1,2,...,N_1^0\}$, there corresponds to at least one terminal node of the transformed sub-tree $\tilde{\Im}^k$ with number n_k such that $\Upsilon_{n_k}^k(u^0) \ge 0$.

The number n_1^0 of corresponding terminal node of tree $\tilde{\Im}$ is derived by formula:

$$n^0 = n_1^0 = n_1 + \sum_{k=2}^{N_1^0}\left((n_k - 1)\cdot\prod_{i=1}^{k-1}\eta_i^1\right). \qquad (18)$$

Recursive formula (18) for node v_t^l, $0 \le l < L - 1$, takes the form:

$$n_t^l = n_{t1}^{l+1} + \sum_{k=2}^{N_1^0}\left((n_{tk}^{l+1} - 1)\cdot\prod_{i=1}^{k-1}\eta_i^{l+1}\right). \qquad (19)$$

For our example shown in Fig. 4, a, applying (18), we have:

$$n_1 = 2, \qquad n_2 = 1, \qquad n_3 = 1, \qquad \eta_1 = 2, \qquad \eta_2 = 1, \qquad \eta_3 = 2,$$

$$n^0 = n_1 + \sum_{k=2}^{3}\left((n_k - 1)\cdot\prod_{i=1}^{k-1}\eta_i^1\right) = n_1 + (n_2 - 1)\cdot\eta_1^1 + (n_3 - 1)\cdot\eta_1^1\cdot\eta_2^1 = 2 + 0 + 0 = 2.$$

Now we may conclude that each of L-level of I-tree $\Im$, in particular $\Im^{\Upsilon}$**,** can be always transformed to the two-level I-tree $\tilde{\Im}^{\Upsilon}$. To this aim we employ the algorithm given above to reduce the number of levels of I-tree $\Im$ stating form the last level L.

Based on formulas (16)–(19) we can also revivify each n_k, $k \in \{1,2,...,N_1^0\}$, by n^0 and generate corresponding system $\Upsilon_{n^0} \ge 0$, $n^0 \in \{1, 2, \ldots, \eta^0\}$, where η^0 is derived by formula of the form (13) or (14).

The technique is applied for generating subregion $W_s \subset W$, $s \in \{1, 2, \ldots, \eta{=}\eta^0\}$ by given point $u^0 \in W$ so that $W_s = \{u \in R^{\sigma} : \Upsilon_s(u_0) \ge 0\}$.

Construction of solution tree

The solution tree $\Im^{\Upsilon}$ describes feasible region W defined by (7) and is constructed as follows. The tree root corresponds to inequality $\varphi = \min\{\varphi_t, t = 1, \ldots, M\} \ge 0$, where $\varphi_t \in \{f\}$. On the first level of $\Im^{\Upsilon}$ we have $\tau_1 = \eta_1$ of nodes, where η_1 is the number of terminal nodes of basic phi-tree $\tilde{\Im}_1$ describing $\Phi_1 \ge 0$, where $\Phi_1 = \max\limits_{i=1,..,\eta_1} f_i^1$, $f_i^1 = \min\limits_{j=1,...,J_i^1} f_{ij}^1$ according to (15). To each node of the first level there corresponds an inequality system $\{\varphi \ge 0, f_{i_1}^1 \ge 0$. To construct the second level of $\Im^{\Upsilon}$ we add η_2 terminal nodes of basic phi-tree $\tilde{\Im}_2$ describing $\Phi_2 \ge 0$ to each node of the first level, where $\Phi_2 = \max\limits_{i=1,..,\eta_2} f_i^2$, $f_i^2 = \min\limits_{j=1,...,J_i^2} f_{ij}^2$. The number of nodes of the second level of $\Im^{\Upsilon}$ becomes $\tau_2 = \eta_1{\cdot}\eta_2$. To each node of the second level there corresponds an inequality system $\{\varphi \ge 0, f_{i_1}^1 \ge 0, f_{i_2}^2 \ge 0$. To construct the k-level of $\Im^{\Upsilon}$ we add η_k terminal nodes of basic phi-tree $\tilde{\Im}_k$ describing $\Phi_k \ge 0$, $\Phi_k = \max\limits_{i=1,..,\eta_k} f_i^k$, $f_i^k = \min\limits_{j=1,...,J_i^k} f_{ij}^k$ to each node of the (k-1)-level of $\Im^{\Upsilon}$. The number of nodes of the k-level of $\Im^{\Upsilon}$ becomes $\tau_k = \eta_1{\cdot}\eta_2{\cdot}\ldots{\cdot}\eta_k$. To each terminal nodes of $\Im_k^{\Upsilon}$ there corresponds an inequality system $\{\varphi \ge 0, f_{i_1}^1 \ge 0, f_{i_2}^2 \ge 0, ..., f_{i_k}^k \ge 0$. Note that $\tau_n = \eta_1{\cdot}\eta_2{\cdot}\ldots{\cdot}\eta_{n-1}{\cdot}\eta_n = \eta$, where η is the number of terminal nodes of $\Im^{\Upsilon}$. Now we may present feasible region W as a union of subregions W_s $s = 1, 2, \ldots, \eta$; see (8). Each W_s corresponds to s-th terminal node of $\Im^{\Upsilon}$ and therefore W_s is determined by an inequality system of the form $\{\varphi \ge 0, f_{s_k}^k \ge 0, k = 1, ..., n$.

Evaluation of the number η of terminal nodes of the solution tree $\Im^{\Upsilon}$

The number η of terminal nodes of the solution tree $\Im^{\Upsilon}$ for problem (1)–(2) depends on the number η' of terminal nodes of phi-tree $\Im_{AB}$ for $\Phi'_{\tau} \geq 0$, τ = 1, 2, …, λ, and the number η'' of the terminal nodes of phi-tree $\Im_{A\Omega^*}$ for $\Phi''_{i} \geq 0$, i = 1, 2, …, N.

We denote the upper estimation of the number of terminal nodes of $\Im_{AB}$ by η'^{*}, and the upper estimation of the number of terminal nodes of $\Im_{A\Omega^*}$ by η''^{*}. Then the upper estimation η^{*}_{N} of the number of terminal nodes of the solution tree $\Im^{\Upsilon}$ for problem (10)–(11) is defined as

$$\eta^{*}_{N} = (\eta'^{*})^{\lambda} \cdot (\eta''^{*})^{N} . \tag{20}$$

Let us derive η'^{*} and η''^{*} in (20).

Let us consider phi-functions for describing non-overlapping and containment constraints

$$\Phi^{AB} = \min\{\Phi'_{k}, k = 1,\dots,n' = n_A \cdot n_B\}, \qquad \Phi^{\Omega^* A} = \min\{\Phi''_{k}, k = 1,2,\dots,n'' = n_A\},$$

where Φ'_{k} is a basic phi-function in Φ^{AB}, Φ''_{k} is a basic phi-function in $\Phi^{\Omega^* A}$. Here Φ^{AB} is a phi-function of objects A and B for non-overlapping constraint; $\Phi^{\Omega^* A}$ is a phi-function of objects A and Ω^* for containment constraint.

We choose here a pair of objects A and B such that $n' = \max\{n'_{\tau}, \tau = 1,\dots,\lambda\}$, where n'_{τ} is the number of basic phi-functions for Φ'_{τ} in (2) and $n'' = \max\{n''_{i}, i = 1,\dots,N\}$, where n''_{i} is the number of basic phi-functions for Φ''_{i} in (2).

We define the number of terminal nodes of $\Im_{AB}$ and $\Im_{A\Omega^*}$ in the form

$$\eta' = \eta'_{1} \cdot \eta'_{2} \cdot \ldots \cdot \eta'_{n'}, \qquad \eta'' = \eta''_{1} \cdot \eta''_{2} \cdot \ldots \cdot \eta''_{n''}, \tag{21}$$

respectively.

Based on (21) we have

$$\eta'^{*} = (\max\{\eta'_{k}, k = 1,2,\dots,n'\})^{n'}, \qquad \eta''^{*} = (\max\{\eta''_{k}, k = 1,2,\dots,n''\})^{n''} . \tag{22}$$

For constructing the solution space of problem (1)–(2) we involve ready-to use free radical basic phi-functions Φ^{AB}, for non-overlapping constraints, $A \in \Re$, $B \in \Re$, and phi-functions $\Phi^{A\Omega^*}$ for containment constraints, $A \in \Re$. Here $\Re$ is a collection of basic objects of four types (see [16] for details).

The values of upper estimations η'^{*} and η''^{*} (22) for optimal packing problem of a pair of composed objects into a rectangular container have been obtained in [17]. In our problem we have $\eta'^{*} = \left(\max\{385,\ 2(2m^2 + 6m + 7)\}\right)^{n'}$, $\eta''^{*} = 16^{n''}$, where m is the maximal number of frontier elements of basic objects which form our composed objects.

In general case to solve problem (1)–(2) by inspecting all terminal nodes η of the solution tree is an unrealistic task, because in fact we have to solve optimally all subproblems (11) of problem (10)-(11) to get global solution. Therefore, we propose an approach to get "good" local optimal solutions of problem (1)-(2) using special optimization procedure involving the algorithm of generating a non-empty subregion $W_s \subseteq W$ by starting point $u^0 \in W$.

Generation of non-empty subregion W_s by starting point $u^0 \in W$

Our aim is to extract from $\Upsilon \geq 0$ an inequality $\Upsilon_s(u) \geq 0$, which describes subregion $W_s \subset W$, such that $u^0 \in W_s = \{u \in R^{\sigma} : \Upsilon_s(u_0) \geq 0\}$, where $\Upsilon_s(u)$ is defined by (9).

We form subregion W_s as follows. We realise an exhaustive search of nodes v^1_s, s = 1, …, η_1, of the first level of $\Im$ sequentially and search for s_1 such that $f^1_{s_1}(u^0) = f^1(u^0) = \max\{f^1_1(u^0), f^1_2(u^0),\dots, f^1_{\eta_1}(u^0)\}$. Then we realise an exhaustive search of offsprings v^2_s , s = 1, …, η_2, of node $v^1_{s_1}$ and search for s_2 such that $f^2_{s_2}(u^0) = f^2(u^0) = \max\{f^2_1(u^0), f^2_2(u^0),\dots, f^2_{\eta_1}(u^0)\}$. And so on.

On the n^{th} level of our solution tree $\Im$ we realise an exhaustive search of nodes v_s^n, $s = 1, \ldots, \eta_n$ which are offsprings of node $v_{s_{n-1}}^{n-1}$ and search for s_n such that $f_{s_n}^n(u^0) = f^n(u^0) = \max\{f_1^n(u^0), f_2^n(u^0),\ldots, f_{\eta_n}^n(u^0)\}$. Then we form inequality system which corresponds to s^{th} terminal node of our solution tree $\Im$ in the form: $W_s = \{u \in R^\sigma : \varphi \geq 0, f_{s_1}^1 \geq 0, f_{s_2}^2 \geq 0,\ldots, f_{s_n}^n \geq 0\}$. To each sequence of numbers $s_1, s_2, \ldots, s_k, \ldots, s_n$ there corresponds the number s.

In Section 5 we give an example of constructing a solution tree. We show how to find by the given starting point u^0 the number n^0 of a system which describes a subregion W_s and corresponds to the terminal node of the transformed tree $\tilde{\Im}$ and associated with $\Upsilon_s(u) \geq 0$.

Solution algorithm

In order to solve problem (1)–(2) defined in Section 2, we propose the algorithm which works very fast and uses multistart method for a set of feasible starting points. For each starting point we apply special algorithm to search for locally optimal solutions. We apply the algorithm introduced in [20] describe it below.

The algorithm involves of the following procedures: 1) generation of a number of starting points from feasible region of problem (1)–(2), employing the starting point algorithm [18]. The algorithm also allows us to fill holes of composed objects by smaller objects. Assuming that each smaller object fixed within the appropriate composed object we further deal with irregular object bounded by one outer counter; 2) search for a local minimum of problem (1)–(2) based on our solution tree technique and employing the algorithm of Local Optimisation Reduction Algorithm (LORA) for each starting point; 3) choice of the best of local minima obtained at the second step as an approximation to the global solution of problem (1)–(2). We develop special solver for layout problems which uses the core representation of inequlities in a sybmol form and provides exact calculation of Jacobian and Hessian matrixes. The search for local minima of nonlinear programming problems is performed by IPOPT algorithm [19].

An essential part of our local optimisation scheme is LORA algorithm that simplifies description of feasible region of the problem and reduces the runtime of local optimisation. It is due to this reduction our strategy can work efficiently with collections of composed objects.

For each starting point $u^0 \in W$ we apply the following local optimisation iterative procedure.

The main idea of the procedure is as follows.

First for each object $T_i(u_i^0)$, $i \in \{1, 2, \ldots, N\}$, we construct minimal enveloping rectangle R_i^0 with sides parallel to axes of fixed coordinate system, here $u_i^0 = (v_i^0, \theta_i^0) = (x_i^0, y_i^0, \theta_i^0)$. Then we extend semisides of R_i^0 by $0.5\rho^-$ and get rectangle R_i.

We assume that the eigen coordinate system of R_i coincides with the eigen coordinate system of T_i. We note that vertices of R_i are defined as p_{iq}, $q = 1, 2, 3, 4$.

We suppose that rectangle $R_i(u_i)$ (conjoint with object $T_i(u_i)$) may move such that each vertex $p_{iq}(u_i) = (x_{iq}(u_i), y_{iq}(u_i))$ has to be arranged within the fixed “square container” $\Omega_{iq}(p_{iq}^0)$ with center point $p_{iq}^0 = p_{iq}(u_i^0) = (x_{iq}^0, y_{iq}^0)$ and side of length δ, i. e

$$p_{iq}(u_i) \in \Omega_{iq}(p_{iq}^0), \quad q = 1, 2, 3, 4. \tag{23}$$

Here δ is a given step of LORA algorithm which calculated depending on sizes of objects.

Thus, relation (23) provides such placement of rotated and translated object $T_i(u_i)$ that any point of $T_i(u_i)$ can vary within δ-square only. The additional constraints on placement parameters u_i we call δ-inequalities involving 16 nonlinear inequalities of the following form:

$$x_{iq}^0 - 0.5\delta \leq x_{iq}(u_i), \qquad x_{iq}(u_i) \leq x_{iq}^0 + 0.5\delta, \qquad y_{iq}^0 - 0.5\delta \leq y_{iq}(u_i), \qquad y_{iq}(u_i) \leq y_{iq}^0 + 0.5\delta,$$

$$p_{iq}(u_i) = v_i + (p_{iq}^0 - v_i^0) \cdot M(-\theta_i^0) \cdot M(\theta_i), \qquad q = 1,2,3,4,$$

where $M(\bullet)$is a rotation matrix, $v_i = (x_i, y_i)$, $v_i^0 = (x_i^0, y_i^0)$. Further, we denote the inequality system of δ-inequalities for all objects $T_i(u_i)$, $i = 1, 2, \ldots, N$ by $\Delta \geq 0$.

By analogy we construct minimal enveloping rectangle R_i^δ of $\Omega_{iq}(p_{iq}^0)$, $q = 1, 2, 3, 4$.

These δ-inequalities provides a motion of object $T_i(u_i)$ within R_i^δ taking into account distance constraint.

Then we construct minimal enveloping rectangle R_{ik}^δ for each basic object $T_i^k(u_i)$, which form composed object $T_i(u_i) = \bigcup_{k=1}^{n_i} T_i^k(u_i)$.

Let us consider now a pair of objects $T_i(u_i)$ and $T_j(u_j)$, $i < j \in I_N$. If $\operatorname{int} R_i^\delta \bigcap \operatorname{int} R_j^\delta = \varnothing$, then we replace phi-inequalities for $T_i(u_i)$ and $T_j(u_j)$, which take part in describing of our feasible region W, by δ-inequalities for placement parameters u_i and u_j. We realize the same transformations for basic objects.

Step 1. For each object $T_i(u_i^0)$ and each its basic object $T_i^k(u_i^0)$ we construct R_i, Ω_{iq}, $q = 1, 2, 3, 4$, R_i^δ and R_{ik}^δ, $k = 1, 2, \ldots, n_i$, $i = 1, 2, \ldots, N$ and form the inequality system $\Delta \geq 0$. Further we note the system of inequalities $\Delta \geq 0$, $\varphi \geq 0$ by $\varphi_\delta \geq 0$.

Step 2. We construct solution tree $\Im_\delta^0$ eliminating such levels for which:

a) $\operatorname{int} R_i^\delta \bigcap \operatorname{int} \Omega^* = \varnothing$ or $\operatorname{int} R_{ik}^\delta \bigcap \operatorname{int} \Omega^* = \varnothing$ for $i \in I_N$, $k \in I_{n_i}$.

b) $\operatorname{int} R_i^\delta \bigcap \operatorname{int} R_j^\delta = \varnothing$ or $\operatorname{int} R_{ik}^\delta \bigcap \operatorname{int} R_{jl}^\delta = \varnothing$ or $\widehat{\Phi}_{ij}^{kl}(u_i^0, u_j^0) \geq 0$, for $i < j \in I_N$, $k \in I_{n_i}$, $l \in I_{n_j}$, $\widehat{\Phi}_{ij}^{kl}$ is a adjusted phi-function for the pair of basic objects $T_i^k(u_i)$ and $T_j^l(u_j)$. We set here that $\widehat{\Phi}_{ij}^{kl} \geq 0$ if $\operatorname{dist}(T_i^k, T_j^l) \geq \rho^- + \sqrt{2}\delta$.

Step 3. Based on $\Im_\delta^0$ we generate the inequality system $\Upsilon_s^0(u) \geq 0$ provided that inequality system $\varphi \geq 0$ is replaced by $\varphi_\delta \geq 0$.

Step 4. Search for a point of local minimum u^{0*} of subproblem

$$\min \kappa(u) \quad \text{s. t.} \quad u \in W_{s_0}^0 \subset R^\sigma,$$

starting from $u^0 \in W_{s^0}^0$, where subregion $W_{s^0}^0 \subset W^0$is described by inequality system $\Upsilon_s^0(u) \geq 0$.

We take point $u^1 = u^{0*}$ as a new starting point, follow steps 1)–3) and form subregion W_{s_1}. Then our algorithm searches for a point of local minimum u^{1*} of subproblem

$$\min \kappa(u) \quad \text{s. t.} \quad u \in W_{s_1}^1 \subset R^\sigma.$$

We take point $u^2 = u^{1*}$ as a starting point for further local optimisation following steps 1)–5).

We repeat the iterative procedure until $\kappa(u^{k*}) = \kappa(u^{(k+1)*})$, where $k = 1, 2, \ldots$ is the number of our iteration procedure. Then point u^{k*} is considered as a point $u^*$of local minimum of problem (1)–(2).

The use of the algorithm allows us to reduce considerably the number of phi-inequalities describing the solution space for local optimisation, which may be crucial even when the number of composed objects $n = 2$ (see [17] for details).

So, while there are $O(m^2)$ pairs of basic objects in the container, our algorithm may in most cases only actively controls $O(m)$ pairs of basic objects (this depends on the sizes of basic objects and the value of δ), because for each basic object only its "δ-neighbors" have to be monitored. Here $m = \sum_{i=`}^{N} n_i$ is the number of all basic objects, n_i is the number of basic objects in object T_i.

The δ parameter provides a balance between the reducing number of inequalities in each NLP subproblem and the number of the subproblems which we need to generate (it also takes computational resources) and solve in order to get a local optimal solution of problem (1)–(2). Our algorithm allows us to reduce considerably computational costs (computational time and memory).

Thus the algorithm reduces the problem (10)–(11) with $O(m^2)$ inequalities describing solution space W to a sequence of subproblems, each with $O(m)$ inequalities describing solution subregion W_s. This reduction is of a paramount importance, since we deal with nonlinear optimization problems.

Illustrative example

For the sake of simplicity we consider tree simple convex objects:

- circle C_1 of radius $r = 5$ with the center point at (0,0);
- polygon K_2 with vertices (0.0, 0.0), (11.0, 0.0), (11.0, 11.0);
- polygon K_3 with vertices (0.0. 0.0), (8.0, 0.0), (14.0, 7.0), (14.0, 15.0),

and rectangular container Ω of width $w = 22$ and variable length l.

We show here how we construct a system of inequality with smooth functions, which describe subregion $W_s \subset W$ by given starting point $u_0 \in W$ and search for a point of local minimum of problem $\min l$ s. t. $u \in W_s$. Here $u = (l, x_1, y_1, x_2, y_2, \theta_2, x_3, y_3, \theta_3)$.

Let starting point $u_0 \in W$ be found, say $l^0 = 30$, $x1^0 = 5.500000$, $y1^0 = 9.959309$, $x2^0 = 4.238658$, $y2^0 = 1.264635$, $\theta2^0 = 0.087266$, $x3^0 = 20.661464$, $y3^0 = 13.080639$, $\theta3^0 = -1.047198$.

Then we form inequality system $W_s \subset W$, such that $u \in W_s$.

First we derive phi-functions for containment constraints:

$\Phi^{C_1\Omega^*} = f^1 = \min\{-x1 + l-5, -y1 + 17, x1 - 5, y1 - 5\};$

$\Phi^{K_2\Omega^*} = f^2 = \min\{-x2 + l, -11*\cos\theta2 - x2 + l, -11*\sin\theta2 - 11*\cos\theta2 - x2 + l, -y2 + 22, 11*\sin\theta2 - y2 + 22, 11*\sin\theta2 - 11*\cos\theta2 - y2 + 22, x2, 11*\cos\theta2 + x2, 11*\sin\theta2 + 11*\cos\theta2 + x2, y2, -11*\sin\theta2 + y2, -11*\sin\theta2 + 11*\cos\theta2 + y2\};$

$\Phi^{K_3\Omega^*} = f^3 = \min\{x3, 8*\cos\theta3 + x3, 7*\sin\theta3 + 14*\cos\theta3 + x3, 15*\sin\theta3 + 14*\cos\theta3 + x3, y3, -8*\sin\theta3 + y3, -14*\sin\theta3 + 7*\cos\theta3 + y3, -14*\sin\theta3 + 15*\cos\theta3 + y3, -x3 + l, -8*\cos\theta3 - x3 + l, -7*\sin\theta3 - 14*\cos\theta3 - x3 + l, -15*\sin\theta3 - 14*\cos\theta3 - x3 + l, -y3 + 22, 8*\sin\theta3 - y3 + 22, 14*\sin\theta3 - 7*\cos\theta3 - y3 + 22, 14*\sin\theta3 - 15*\cos\theta3 - y3 + 22\}.$

Each phi-tree for containment constraints $f^1 \geq 0, f^2 \geq 0, f^3 \geq 0$ has only one node. Therefore we introduce $f_0 = \min\{f^1, f^2, f^3\} \geq 0$, for which I-tree also has only one terminal node, i. e. $\eta^0 = 1$.

Then we define phi-functions for non-overlapping constraints:

$\Phi^{C_1K_2} = f^{12} = \max\{f_1^{12}, f_2^{12}, f_3^{12}, f_4^{12}, f_5^{12}, f_6^{12}\}$, where

$f_1^{12} = -\sin\theta2*x1 - \cos\theta2*y1 + x2*\sin\theta2 + y2*\cos\theta2 - 5;$

$f_2^{12} = \min\{x2^2 - 2*x1*x2 + x1^2 + y2^2 - 2*y1*y2 + y1^2 - 25, (-0.3827*\sin\theta2 - 0.9239*\cos\theta2)*x1 + (0.9239*\sin\theta2 - 0.3827*\cos\theta2)*y1 - (-0.3827*\sin\theta2 - 0.9239*\cos\theta2)*x2 - (0.9239*\sin\theta2 - 0.3827*\cos\theta2)*y2 - 1.9134\};$

$f_3^{12} = \cos\theta2*x1 - \sin\theta2*y1 - x2*\cos\theta2 + y2*\sin\theta2 - 16;$

$f_4^{12} = \min\{(11*\cos\theta2 + x2)^2 - 2*(11*\cos\theta2 + x2)*x1 + x1^2 + (-11*\sin\theta2 + y2)^2 - 2*(-11*\sin\theta2 + y2)*y1 + y1^2 - 25, (-0.7071*\sin\theta2 + 0.7071*\cos\theta2)*x1 + (-0.7071*\sin\theta2 - 0.7071*\cos\theta2)*y1 - (-0.7071*\sin\theta2 + 0.7071*\cos\theta2)*x2 - (-0.7071*\sin\theta2 - 0.7071*\cos\theta2)*y2 - 11.3137\};$

$f_5^{12} = (0.7071*\sin\theta2 - 0.7071*\cos\theta2)*x1 + (0.7071*\sin\theta2 + 0.7071*\cos\theta2)*y1 - (0.7071*\sin\theta2 - 0.7071*\cos\theta2)*x2 - (0.7071*\sin\theta2 + 0.7071*\cos\theta2)*y2 - 5;$

$f_6^{12} = \min\{(11*\sin\theta2 + 11*\cos\theta2 + x2)^2 - 2*(11*\sin\theta2 + 11*\cos\theta2 + x2)*x1 + x1^2 + (-11*\sin\theta2 + 11*\cos\theta2 + y2)^2 - 2*(-11*\sin\theta2 + 11*\cos\theta2 + y2)*y1 + y1^2 - 25, (0.9239*\sin\theta2 + 0.3827*\cos\theta2)*x1 + (-0.3827*\sin\theta2 + 0.9239*\cos\theta2)*y1 - (0.9239*\sin\theta2 + 0.3827*\cos\theta2)*x2 - (-0.3827*\sin\theta2 + 0.9239*\cos\theta2)*y2 - 16.2856\}.$

Transformed phi-tree for $f^{12} \geq 0$ has $\eta^{12} = 6$ terminal nodes (Fig. 5).

$\Phi^{C_1K_3} = f^{13} = \max\{f_1^{13}, f_2^{13}, f_3^{13}, f_4^{13}, f_5^{13}, f_6^{13}, f_7^{13}, f_8^{13}\}$, where

$f_1^{13} = -\sin\theta3*x1 - \cos\theta3*y1 + x3*\sin\theta3 + y3*\cos\theta3 - 5$;

$f_2^{13} = \min\ \{x3^2 - 2*x1*x3 + x1^2 + y3^2 - 2*y1*y3 + y1^2 - 25, (-0.3985*\sin\theta3 - 0.9171*\cos\theta3)*x1 + (0.9171*\sin\theta3 - 0.3985*\cos\theta3)*y1 - (-0.3985*\sin\theta3 - 0.9171*\cos\theta3)*x3 - (0.9171*\sin\theta3 - 0.3985*\cos\theta3)*y3 - 1.9927\}$;

$f_3^{13} = (-0.6508*\sin\theta3 + 0.7593*\cos\theta3)*x1 + (-0.7593*\sin\theta3 - 0.6508*\cos\theta3)*y1 - (-0.6508*\sin\theta3 + 0.7593*\cos\theta3)*x3 - (-0.7593*\sin\theta3 - 0.6508*\cos\theta3)*y3 - 11.0741$;

$f_4^{13} = (0.6823*\sin\theta3 - 0.7311*\cos\theta3)*x1 + (0.7311*\sin\theta3 + 0.6823*\cos\theta3)*y1 - (0.6823*\sin\theta3 - 0.7311*\cos\theta3)*x3 - (0.7311*\sin\theta3 + 0.6823*\cos\theta3)*y3 - 5$;

$f_5^{13} = \cos\theta3*x1 + (-\sin\theta3)*y1 - x3*\cos\theta3 + y3*\sin\theta3 - 19$;

$f_6^{13} = \min\ \{(7*\sin\theta3 + 14*\cos\theta3 + x3)^2 - 2*(7*\sin\theta3 + 14*\cos\theta3 + x3)*x1 + x1^2 + (-14*\sin\theta3 + 7*\cos\theta3 + y3)^2 - 2*(-14*\sin\theta3 + 7*\cos\theta3 + y3)*y1 + y1^2 - 25, (-0.3469*\sin\theta3 + 0.9379*\cos\theta3)*x1 + (-0.9379*\sin\theta3 - 0.3469*\cos\theta3)*y1 - (-0.3469*\sin\theta3 + 0.9379*\cos\theta3)*x3 - (0.9379*\sin\theta3 - 0.3469*\cos\theta3)*y3 - 15.3912\}$;

$f_7^{13} = \min\ \{(8*\cos\theta3 + x3)^2 - 2*(8*\cos\theta3 + x3)*x1 + x1^2 + (-8*\sin\theta3 + y3)^2 - 2*(-8*\sin\theta3 + y3)*y1 + y1^2 - 25, (-0.9085*\sin\theta3 + 0.4179*\cos\theta3)*x1 + (-0.4179*\sin\theta3 - 0.9085*\cos\theta3)*y1 - (-0.9085*\sin\theta3 + 0.4179*\cos\theta3)*x3 - (-0.4179*\sin\theta3 - 0.9085*\cos\theta3)*y3 - 7.8855\}$;

$f_8^{13} = \min\ \{(15*\sin\theta3 + 14*\cos\theta3 + x3)^2 - 2*(15*\sin\theta3 + 14*\cos\theta3 + x3)*x1 + x1^2 + (-14*\sin\theta3 + 15*\cos\theta3 + y3)^2 - 2*(-14*\sin\theta3 + 15*\cos\theta3 + y3)*y1 + y1^2 - 25, (0.9303*\sin\theta3 + 0.3667*\cos\theta3)*x1 + (-0.3667*\sin\theta3 + 0.9303*\cos\theta3)*y1 - (0.9303*\sin\theta3 + 0.3667*\cos\theta3)*x3 - (-0.3667*\sin\theta3 + 0.9303*\cos\theta3)*y3 - 20.9224\}$.

Transformed phi-tree for $f^{13} \geq 0$ has $\eta^{13} = 8$ terminal nodes (Fig. 5).

$\Phi^{K_2K_3} = f^{23} = \max\{f_1^{23}, f_2^{23}, f_3^{23}, f_4^{23}, f_5^{23}, f_6^{23}, f_7^{23}\}$, where

$f_1^{23} = \min\ \{-\sin\theta2*x3 - \cos\theta2*y3 + x2*\sin\theta2 + y2*\cos\theta2, -\sin\theta2*(8*\cos\theta3 + x3) - \cos\theta2*(-8*\sin\theta3 + y3) + x2*\sin\theta2 + y2*\cos\theta2, -\sin\theta2*(7*\sin\theta3 + 14*\cos\theta3 + x3) - \cos\theta2*(-14*\sin\theta3 + 7*\cos\theta3 + y3) + x2*\sin\theta2 + y2*\cos\theta2, -\sin\theta2*(15*\sin\theta3 + 14*\cos\theta3 + x3) - \cos\theta2*(-14*\sin\theta3 + 15*\cos\theta3 + y3) + x2*\sin\theta2 + y2*\cos\theta2\}$;

$f_2^{23} = \min\ \{\cos\theta2*x3 + (-\sin\theta2)*y3 - x2*\cos\theta2 + y2*\sin\theta2 - 11, \cos\theta2*(8*\cos\theta3 + x3) + (-\sin\theta2)*(-8*\sin\theta3 + y3) - x2*\cos\theta2 + y2*\sin\theta2 - 11, \cos\theta2*(7*\sin\theta3 + 14*\cos\theta3 + x3) + (-\sin\theta2)*(-14*\sin\theta3 + 7*\cos\theta3 + y3) - x2*\cos\theta2 + y2*\sin\theta2 - 11, \cos\theta2*(15*\sin\theta3 + 14*\cos\theta3 + x3) + (-\sin\theta2)*(-14*\sin\theta3 + 15*\cos\theta3 + y3) - x2*\cos\theta2 + y2*\sin\theta2 - 11\}$;

$f_3^{23} = \min\ \{(0.7071*\sin\theta2 - 0.7071*\cos\theta2)*x3 + (0.7071*\sin\theta2 + 0.7071*\cos\theta2)*y3 - (0.7071*\sin\theta2 - 0.7071*\cos\theta2)*x2 - (0.7071*\sin\theta2 + 0.7071*\cos\theta2)*y2, (0.7071*\sin\theta2 - 0.7071*\cos\theta2)*(8*\cos\theta3 + x3) + (0.7071*\sin\theta2 + 0.7071*\cos\theta2)*(-8*\sin\theta3 + y3) - (0.7071*\sin\theta2 - 0.7071*\cos\theta2)*x2 - (0.7071*\sin\theta2 + 0.7071*\cos\theta2)*y2, (0.7071*\sin\theta2 - 0.7071*\cos\theta2)*(7*\sin\theta3 + 14*\cos\theta3 + x3) + (0.7071*\sin\theta2 + 0.7071*\cos\theta2)*(-14*\sin\theta3 + 7*\cos\theta3 + y3) - (0.7071*\sin\theta2 - 0.7071*\cos\theta2)*x2 - (0.7071*\sin\theta2 + 0.7071*\cos\theta2)*y2, (0.7071*\sin\theta2 - 0.7071*\cos\theta2)*(15*\sin\theta3 + 14*\cos\theta3 + x3) + (0.7071*\sin\theta2 + 0.7071*\cos\theta2)*(-14*\sin\theta3 + 15*\cos\theta3 + y3) - (0.7071*\sin\theta2 - 0.7071*\cos\theta2)*x2 - (0.7071*\sin\theta2 + 0.7071*\cos\theta2)*y2\}$;

$f_4^{23} = \min\ \{-\sin\theta3*x2 - \cos\theta3*y2 + x3*\sin\theta3 + y3*\cos\theta3, -\sin\theta3*(11*\cos\theta2 + x2) - \cos\theta3*(-11*\sin\theta2 + y2) + x3*\sin\theta3 + y3*\cos\theta3, -\sin\theta3*(11*\sin\theta2 + 11*\cos\theta2 + x2) - \cos\theta3*(-11*\sin\theta2 + 11*\cos\theta2 + y2) + x3*\sin\theta3 + y3*\cos\theta3\}$;

$f_5^{23} = \min\{(-0.6508*\sin\theta3 + 0.7593*\cos\theta3)*x2 + (-0.7593*\sin\theta3 - 0.6508*\cos\theta3)*y2 - (-0.6508*\sin\theta3 + 0.7593*\cos\theta3)*x3 - (-0.7593*\sin\theta3 - 0.6508*\cos\theta3)*y3 - 6.0741, (-0.6508*\sin\theta3 + 0.7593*\cos\theta3)*(11*\cos\theta2 + x2) + (-0.7593*\sin\theta3 - 0.6508*\cos\theta3)*(-11*\sin\theta2 + y2) - (-0.6508*\sin\theta3 + 0.7593*\cos\theta3)*x3 - (-0.7593*\sin\theta3 - 0.6508*\cos\theta3)*y3 - 6.0741, (-0.6508*\sin\theta3 + 0.7593*\cos\theta3)*(11*\sin\theta2 + 11*\cos\theta2 + x2) + (-0.7593*\sin\theta3 - 0.6508*\cos\theta3)*(-11*\sin\theta2 + 11*\cos\theta2 + y2) - (-0.6508*\sin\theta3 + 0.7593*\cos\theta3)*x3 - (-0.7593*\sin\theta3 - 0.6508*\cos\theta3)*y3 - 6.0741\}$;

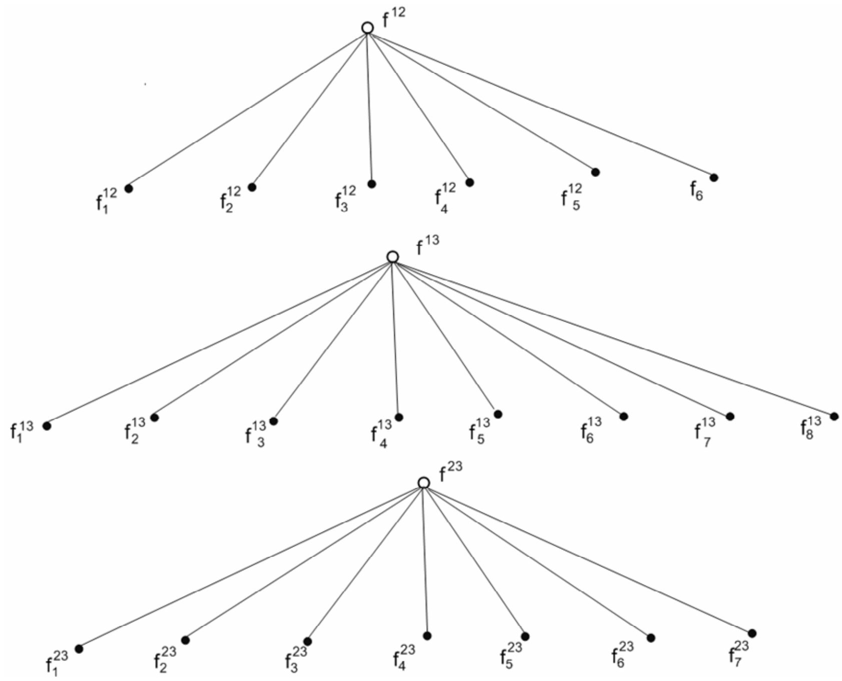


Fig. 5. Tranformed phi-trees for containment constraints for $f^{12} \geq 0, f^{13} \geq 0, f^{23} \geq 0$

f_6^{23} = min {cosθ3*x2 + (–sinθ3)*y2 – x3*cosθ3 + y3*sinθ3 – 14, cosθ3*(11*cosθ2 + x2) + (–sinθ3)*(–11*sinθ2 + y2) – x3*cosθ3 + y3*sinθ3 – 14, cosθ3*(11*sinθ2 + 11*cosθ2 + x2) + (–sinθ3)* (–11*sinθ2 + 11*cosθ2 + y2) – x3*cosθ3 + y3*sinθ3 – 14};

f_7^{23} = min {(0.6823*sinθ3 – 0.7311*cosθ3)*x2 + (0.7311*sinθ3 + 0.6823*cosθ3)*y2 – (0.6823*sinθ3 – 0.7311*cosθ3)*x3 – (0.7311*sinθ3 + 0.6823*cosθ3)*y3,(0.6823*sinθ3 – 0.7311*cosθ3)* (11*cosθ2 + x2) + (0.7311*sinθ3 + 0.6823*cosθ3)*(–11*sinθ2 + y2) – (0.6823*sinθ3 – 0.7311*cosθ3)*x3 – (0.7311*sinθ3 + 0.6823*cosθ3)*y3, (0.6823*sinθ3 – 0.7311*cosθ3)*(11*sinθ2 + 11*cosθ2 + x2) + (0.7311*sinθ3 + 0.6823*cosθ3)* (–11*sinθ2 + 11*cosθ2 + y2) – (0.6823*sinθ3 – 0.7311*cosθ3)*x3 – (0.7311*sinθ3 + 0.6823*cosθ3)*y3}.

Transformed phi-tree for $f^{23} \geq 0$ has $\eta^{23} = 7$ (Fig. 5) terminal nodes.

We build the solution tree of our problem using phi-trees. The solution tree has $\eta = \eta^0 \cdot \eta^{12} \cdot \eta^{13} \cdot \eta^{23} = 1 \cdot 6 \cdot 7 \cdot 8 = 336$ terminal nodes.

In order to generate a system of inequalities with smooth functions which describe subregion $W_s \subset W$ we derive values of each our function at point u_0:

$\lambda(u_0) = 0.2950053 \approx 0.3$, since

$f^1(u_0) = \min\{$25.7613416, 14.8031999, 13.8444867, 20.7353650, 21.6940780, 10.7359363, 4.2386584, 4.2386584, 15.1968001, 16.1555133, 16.1555133, 1.2646352, 0.3059220, 11.2640637} = 0.5,

$f^2(u_0) =$ min{25.7613416, 14.8031999, 13.8444867, 20.7353650, 21.6940780, 10.7359363, 4.2386584, 4.2386584, 15.1968001, 16.1555133, 16.1555133, 1.2646352, 0.3059220, 11.2640637} = 0.3059220,

$f^3(u_0) =$ min {20.6614640, 16.6614640, 7.5992862, 0.6710829, 13.0806390, 20.0088422, 21.7049947, 17.7049947, 9.3385360, 13.3385360, 22.4007138, 29.3289171, 8.9193610, 1.9911578, 0.2950053, 4.2950053} = 0.2950053.

$f^{12}(u_0) = 0.8497317 \approx 0.85$, since

$f_1^{12}(u_0) = -13.7715209$, $f_2^{12}(u_0) =$ min {52.1883321, –5.7309188} = –5.7309188,

$f_3^{12}(u_0) = -15.5012489$, $f_4^{12}(u_0) =$ min {162.2158087, –17.1634402} = –17.1634402,

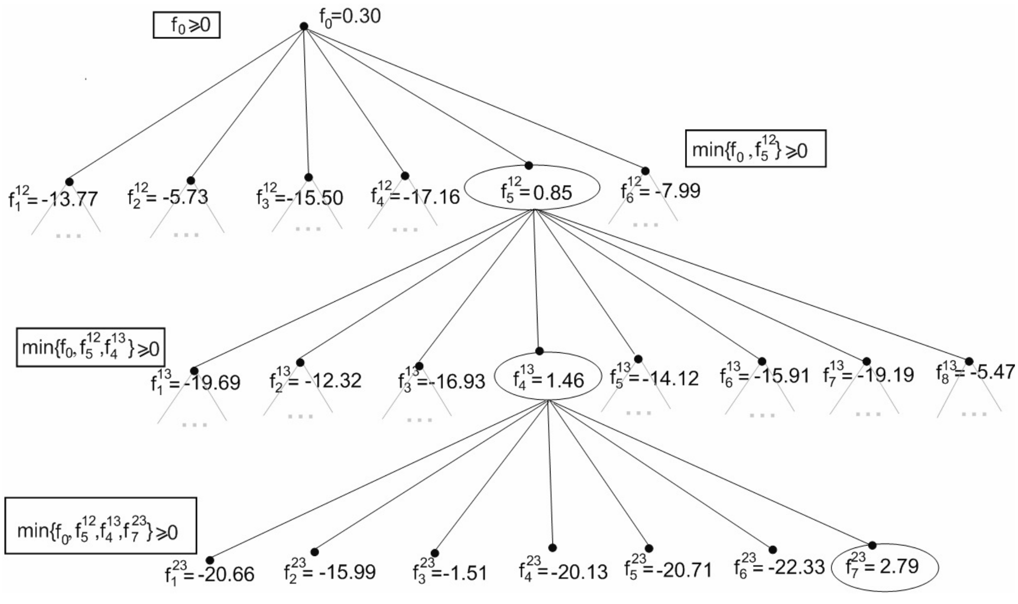


Fig. 6 Generating of an inequality system, using point $u_0 \in W$

$f_5^{12}(u_0) = 0.8497317,\ f_6^{12}(u_0) = \min\ \{\ 90.2423482, -7.9909173\} = -7.9909173.$

$f^{13}(u_0) = 1.4580729\ \approx 1.46$, since

$f_1^{13}(u_0) = -19.6908780,$

$f_2^{13}(u_0) = \min\ \{214.6126916, -12.3212292\} = -12.3212292,$

$f_3^{13}(u_0) = -16.9314140,\ f_4^{13}(u_0) =\ 1.4580729,\ f_5^{13}(u_0) = -14.1224191,$

$f_6^{13}(u_0) = \min\ \{117.3681339, -15.9135265\} = -15.9135265,$

$f_7^{13}(u_0) = \min\ \ \{200.5713968, -19.1941407\} = -19.1941407,$

$f_8^{13}(u_0) = \min\ \ \{58.3140862, -5.4663458\} = -5.4663458,$

$f^{23}(u_0) = 2.7853770\ \approx 2.79$, since

$f_1^{23}(u_0) = \min\ \{-13.2023824, -19.7555987, -20.6554760, -16.0668645\} = -20.6554760,$

$f_2^{23}(u_0) = \min\ \{4.3304797, -0.2581318, -9.4336548, -15.9868711\} = -15.9868711,$

$f_3^{23}(u_0) = \min\ \{-1.5047920, 6.3736700, 13.4980538, 14.8872392\} = -1.5047920,$

$f_4^{23}(u_0) = \min\ \{-20.1305692, -11.1198967, -4.8105559\} = -20.1305692,$

$f_5^{23}(u_0) = \min\ \{-20.7097338, -19.6360746, -8.6885974\} = -20.7097338,$

$f_6^{23}(u_0) = \min\ \{-16.0215566, -22.3308974, -13.3202249\} = -22.3308974,$

$f_7^{23}(u_0) = \min\ \{15.2133244, 13.6776549, 2.7853770\} = 2.7853770.$

Scheme of generating of an inequality system, which describes nonempty subregion $W_s \subset W$, using point $u_0 \in W$, is given in Fig. 6.

$-x1 + 1 - 5^{i\ \ 0}, -y1 + 17^{i\ \ 0}, x1 - 5^{i\ \ 0}, y1 - 5^{i\ \ 0},$

$-x2 + 1^{i\ \ 0},$

$-11{*}\cos\theta2 - x2 + 1^{i\ \ 0},$

$-11{*}\sin\theta2 - 11{*}\cos\theta2 - x2 + 1^{i\ \ 0},$

$-y2 + 22,\ 11{*}\sin\theta2 - y2 + 22^{i\ \ 0},$

$11{*}\sin\theta2 - 11{*}\cos\theta2 - y2 + 22^{i\ \ 0},$

$x2^{i\ \ 0},$

11*cosθ2 + x2 i 0 ,
11*sinθ2 + 11*cosθ2 + x2 i 0 ,
y2 i 0 ,
–11*sinθ2 + y2 i 0 ,
–11*sinθ2 + 11*cosθ2 + y2 i 0 ,
x3 i 0 ,
8*cosθ3 + x3 i 0 ,
7*sinθ3 + 14*cosθ3 + x3 i 0 ,
15*sinθ3 + 14*cosθ3 + x3 i 0 ,
y3 i 0 ,
–8*sinθ3 + y3 i 0 ,
–14*sinθ3 + 7*cosθ3 + y3 i 0 ,
–14* sinθ3 + 15*cosθ3 + y3 i 0 ,
–x3 + l i 0 ,
–8*cosθ3 – x3 + l i 0 ,
–7*sinθ3 – 14*cosθ3 – x3 + l i 0 ,
–15*sinθ3 – 14*cosθ3 – x3 + l i 0 ,
–y3 + 22 i 0 ,
8*sinθ3 – y3 + 22 i 0 ,
14* sinθ3 – 7*cosθ3 – y3 + 22 i 0 ,
14*sinθ3 – 15*cosθ3 – y3 + 22 i 0 ,
(0.7071*sinθ2 – 0.7071*cosθ2)*x1 + (0.7071*sinθ2 + 0.7071*cosθ2)*y1 – (0.7071*sinθ2 –
0.7071*cosθ2)*x2 – (0.7071*sinθ2 + 0.7071*cosθ2)*y2 – 5 i 0 ,
(0.6823*sinθ3 – 0.7311*cosθ3)*x1 + (0.7311*sinθ3 + 0.6823*cosθ3)*y1 – (0.6823*sinθ3 –
0.7311*cosθ3)*x3 – (0.7311*sinθ3 + 0.6823*cosθ3)*y3 – 5 i 0 ,
(0.6823*sinθ3 – 0.7311*cosθ3)*x2 + (0.7311*sinθ3 + 0.6823*cosθ3)*y2 – (0.6823*sinθ3 –
0.7311*cosθ3)*x3 – (0.7311* sinθ3 + 0.6823*cosθ3)*y3 i 0 ,
(0.6823*sinθ3 – 0.7311*cosθ3)*(11*cosθ2 + x2) + (0.7311*sinθ3 + 0.6823*cosθ3)*(–11*sinθ2 + y2) –
(0.6823*sinθ3 – 0.7311*cosθ3)*x3 – (0.7311*sinθ3 + 0.6823*cosθ3)*y3 i 0 ,
(0.6823*sinθ3 – 0.7311*cosθ3)*(11*sinθ2 + 11*cosθ2 + x2) + (0.7311*sinθ3 + 0.6823*cosθ3)*
(–11*sinθ2 + 11*cosθ2 + y2) – (0.6823*sinθ3 – 0.7311*cosθ3)*x3 –
(0.7311*sinθ3 + 0.6823*cosθ3)*y3 i 0 .

We found a local minimum point $u^* = (l^*, x_1^*, y_1^*, x_2^*, y_2^*, \theta_2^*, x_3^*, y_3^*, \theta_3^*)$, where l^*=15.8871253, x_1^*=10.8871253, y_1^*=13.0710679, x_2^*=4.8871253, y_2^*=0.0000000, θ_2^*=0.0000000, x_3^*=6.5463322508, y_3^*=21.9999999, θ_3^*=2.4659396.

Computaional results

We give a number of examples to demonstrate the effectiveness of our methodology for rectangular domain given in [20].

For local optimisation in our programs we use IPOPT, which is available at an open access non-commercial software depository (https://projects.coin-or.org/Ipopt).

We use computer AMD Athlon 64 X2 5200+ for our computational experiments.

The comparison was carried out with the results given in [8] and [9]. The results have been improved (see Table 1).

The Table 1 shows the results of comparison (the length of the occupied parts of the strip) for five data sets of Profile1, Profile2, Profile3, Profile4 and Profile5.

Table 1. Comparison results

Data sets	Profile1	Profile2	Profile3	Profile4	Profile5
The best result given in [8], [9]	1359.90	3194.19	7881.13	2425.26	3332.70
Our results	1318.49	3104.72	7501.96	2382.62	2996.306

Further we applied our method to some instances used in recent paper [21] by Kallrath and Rebennack and compare our optimal solutions to theirs (see Table 2).

Table 2. Comparison of our results to those in [21]

Name	Our result	The best from [21]	Improvement (%)
TC30	95.36535	103.45212	8.4798
TC50	154.470487	166.91505	8.0563
TC100	300.5142183	322.64663	8.3660

Conclusion

We propose here the automatic feasible region generator, using phi-trees. The generator allows us to form ready-to-use systems of inequalities with smooth functions in order to apply efficient nonlinear optimisation procedures. We develop an efficient solution algorithm and original solver for nonsmooth layout problems which uses the core representation of inequlities in a sybmol form and provides exact calculation of Jacobian and Hessian matrixes. The search for local minima of NLP-problems is performed by IPOPT algorithm. An essential part of our local optimisation scheme is LORA algorithm that simplifies description of feasible region of the problem and reduces the runtime of local optimisation. It is due to this reduction our strategy can work efficiently with collections of composed objects and search for “good” local-optimal solutions for layout problems in reasonable time.

Reference

1. Wascher, G. An improved typology of cutting and packing problems / G. Wascher, H. Hauner, H. Schumann // European J. Operational Research. – 2007. – № 183 (3). – P. 1109–1130.
2. Gomes, A. M. Irregular Packing Problems: Industrial Applications and New Directions Using Computational Geometry / A. M. Gomes // IFAC Proc. Volumes. – 2013. – Vol. 46 (7). – P. 378–383.
3. Chazelle, B. The complexity of cutting complexes / B. Chazelle, H. Edelsbrunner, L. J. Guibas // Discr. & Comput. Geom. – 1989. – № 4 (2). – P. 81–139.
4. Bennell, J. A. The geometry of packing problems: A tutorial / J. A. Bennell, J. F. Oliveira // European J. Operational Research. – 2008. – № 184 (2) – P. 397–415.
5. Bennell, J. A. A tutorial in irregular shape packing problem / J. A. Bennell, J. F. Oliveira // Oper. Research Soc. – 2009. – Vol. 60. – P. 93–105.
6. Two-dimensional cutting problem basic complexity results and algorithms for irregular shapes / J. Blazewicz, M. Drozdowski, B. Soniewicki, R. Walkowiak // Found. Cont. Eng. – 1989. – № 14 (4). – P. 137–160.
7. Burke, E. K. Complete and robust no-fit polygon generation for the irregular stock cutting problem / E. K. Burke, R. S. R. Hellier, G. Kendall, G. Whitwell // EJOR – 2007. – № 179 (1). – P. 27–49.
8. Whitwell, G. Novel Heuristic and Metaheuristic Approaches to Cutting and Packing / School of Computer Science and Information Technology // University of Nottingham; – 2005, PhD Thesis. – 313 p.
9. Irregular packing using the line and arc no-fit polygon / E. K. Burke, R. Hellier, G. Kendall, G. Whitwell // Operations Research. –2010. – № 58 (4). – P. 948–970.
10. Milenkovic, V. Rotational polygon containment and minimum enclosure using only robust 2d constructions / V. Milenkovic // Computational Geometry. –1999. – № 13 (1). – P. 3–19.
11. Milenkovic, V. Rotational polygon overlap minimization and compaction / V. Milenkovic // Computational Geometry. –1998. – № 10 (4). – P. 305–318.
12. Circle Covering Representation for Nesting problems with continuous rotations / P. Rocha, R. Rodrigues, A. M. Gomes et al. // Preprints of the 19th World Congress, The International Federation of Automatic Control, Cape Town, South Africa. August 24–29. – 2014.
13. Leung, S. C. H. Extended local search algorithm based on nonlinear programming for two-dimensional irregular strip packing problem / S. C. H. Leung, Y. Lin, D. Zhang // Computers & Operations Research. – 2012. – № 39 (3). – P. 678–686.

14. Tools of mathematical modelling of arbitrary object packing problems/ J. Bennell, G. Scheithauer, Yu. Stoyan, T. Romanova // J. Annals of Operations Research, Publisher Springer Netherlands. – 2010. – № 179(1). – P. 343–368.
15. Chernov, N. Mathematical model and efficient algorithms for object packing problem / N. Chernov, Yu. Stoyan, T. Romanova // Computational Geometry: Theory and Appl. – 2010. – № 43 (5). – P. 533–553.
16. Phi-functions for 2D objects formed by line segments and circular arcs / N. Chernov, Y. Stoyan, T. Romanova, A. Pankratov // Advances in Operations Research. –2012. – 26 p.
17. Optimal clustering of a pair of irregular objects / J. Bennell, G. Scheithauer, Y. Stoyan et al. // J. Global Optimisation. – 2015. –№ 61 (3). – P. 497–524.
18. Pankratov, A. V. Placement of non-convex polygons with rotations into a non-convex polygon / A. V. Pankratov, Yu. G. Stoyan // Proc. Workshop Cutting Stock Problems. – 2005. Alutus, Miercurea-Ciuc, Romania. – P. 29–36.
19. Wachter, A. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming/ A. Wachter, L. T. Biegler // Mathematical Programming. – 2006. – № 106 (1). – P. 25–57.
20. Stoyan, Yu. Cutting and Packing problems for irregular objects with continuous rotations: mathematical modeling and nonlinear optimization / Yu. Stoyan, A. Pankratov, T. Romanova // J. Operational Research Society, – 2016. – № 67 (5). – P. 786–800.
21. Kallrath, J. Cutting Ellipses from Area-Minimizing Rectangles / J. Kallrath, S. Rebennack // J. Global Optimisation. – 2014. – № 59 (2–3). – P. 405–437.

Поступила в редакцию 06.09.16

[1] **І. В. Сергієнко**, акад. НАН України
[2] **О. М. Литвин**, д-р фіз.-мат. наук
[2] **О. О. Литвин**, канд. фіз.-мат. наук
[3] **О. В. Ткаченко**, канд. фіз.-мат. наук
[3] **О. Л. Грицай**

[1] Інститут кібернетики імені В.М. Глушкова НАН України, м. Київ
[2] Українська інженерно-педагогічна академія, м. Харків, e-mail: academ_mail@ukr.net
[2] ДП СКБ «Івченко-Прогрес»,
м. Запоріжжя, e-mail: avt2007@outlook.com

Ключові слова: *збереження класу диференційовності, сліди функції, сліди похідних на лінії, Ермітова інтерлінація.*

УДК 519.6

ПОБУДОВА ТА ДОСЛІДЖЕННЯ ОПЕРАТОРІВ ЕРМІТОВОЇ ІНТЕРЛІНАЦІЇ ФУНКЦІЙ ДВОХ ЗМІННИХ НА СИСТЕМІ НЕПЕРЕТИННИХ ЛІНІЙ ІЗ ЗБЕРЕЖЕННЯМ КЛАСУ ДИФЕРЕНЦІЙОВНОСТІ

Побудовано та досліджено оператори інтерлінації функцій двох змінних із збереженням класу диференційовності, якому належить наближувана функція за умови, що сліди цих операторів і сліди їх частинних похідних за однією із змінних до фіксованого порядку співпадають на заданій системі ліній з відповідними слідами наближуваної функції.

Вступ

Нехай $f^{(0,s)}(x,y)=\dfrac{\partial^s f(x,y)}{\partial y^s}$. Оператори ермітової інтерлінації, що використовують для своєї побудови сліди наближуваної функції та її частинних похідних до заданого порядку $N \ge 1$ на заданій системі паралельних прямих

$$E_{MN}f(x,y)=\sum_{k=1}^{M}\sum_{s=0}^{N} f^{(0,s)}(x,y_k)h_{k,s}(y)\frac{(y-y_k)^s}{s!},$$

$$f^{(0,s)}(x,y_k)=\left.\frac{\partial^s f(x,y)}{\partial y^s}\right|_{y=y_k},$$

$$h_{k,s}^{(p)}(y_\ell)=\delta_{k,\ell}\delta_{p,s}, \qquad p=0,1,\dots,N-s;$$

© І. В. Сергієнко, О. М. Литвин, О. О. Литвин, О. В. Ткаченко, О. Л. Грицай, 2016